***Programa MikroEulerLCDTest***

***Escreve em um LCD 16x2 HD44780:***

**MikroEuler v0.19**

**RevA - 2k of RAM**

***Autor: Edson Jr. Acordi***

***Data: 02/12/2021***

***Versão: v0.1***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Endereço** | **Código** | **Instrução** | **Descrição** | **Comentário** |
| ; ==== Ajusta Endereço de Retorno da Subrotina de Atraso ==== | | | | |
| 000 | 0000 | LDI ACC, 0 |  | ; Carrega ACC com zero (Primeiro Índice de retorno = 0) |
| 001 | 0A0F | STW RAM[0:0:F], ACC |  | ; Salva o índice de retorno na RAM |
| 002 | 300x | ~~LDI RD, Delay\_ms[8:11]~~ |  | ; Carrega o endereço alto da Subrotina em RD |
| ; ==== Ajusta o número total de ciclos da Subrotina Delay\_ms ==== | | | | |
| 003 | 000F | LDI ACC, Fh |  | ; Prepara para ajustar número de ciclos para Delay\_ms |
| 004 | 0A02 | STW RAM[0:0:2], ACC |  | ; Ajusta atraso máximo para Delay2\_ms e Delay3\_ms |
| 005 | 0C64 | JPI Delay\_ms |  | ; Chama a Subrotina Delay\_ms |
| ; **==== Inicialização do LCD ====**  ; OUTA => Data pins D7:D4 of LCD (4-bit mode)  ; OUTB:  ; OUTB[0] = RS (RS = 0, comando – RS = 1, dado)  ; OUTB[1] = E (Pulso de High para low)  ; Obs.: RW is connected at GND.  ; O delay da transição de High para Low do sinal “E” deve ser ajustado conforme o cristal utilizado. Para cristal de 1MHz, uma instrução é  ; suficiente (1 us => SUB ACC, 0) | | | | |
| ; Envia Comando 02 em D7:D4 | | | | |
| **RET\_Delay\_ms\_00:** |  |  |  |  |
| 006 | 0400 | OUTA 0 |  | **; Send Higher Nibble** |
| 007 | 0500 | OUTB 0 |  | ; Prepara para enviar Cmd - OUTB[0] = RS (bit 0 da OUTB) |
| 008 | 0502 | OUTB 2 |  | ; Prepara o sinal “E” - OUTB[1] = E = 1 (bit 1 da OUTB) |
| 009 | 0B00 | SUB ACC, 0 |  | ; NOP |
| 00A | 0500 | OUTB 0 |  | ; Finaliza o comando (“E” – Enable vai de High para Low) |
| 00B | 0001 | LDI ACC, 1 |  | ; Carrega ACC com 1 (Segundo Índice de retorno = 1) |
| 00C | 0A0F | STW RAM[0:0:F], ACC |  | ; Salva o índice de retorno na RAM |
| 00D | 300x | ~~LDI RD, Delay\_ms[8:11]~~ |  | ; Carrega o endereço alto da Subrotina em RD |
| 00E | 0C64 | JPI Delay\_ms |  | ; Chama a Subrotina Delay\_ms |
| **RET\_Delay\_ms\_01:** |  |  |  |  |
| 00F | 0402 | OUTA 2 |  | **; Send Lower Nibble** |
| 010 | 0502 | OUTB 2 |  | ; Prepara o sinal “E” - OUTB[1] = E = 1 (bit 1 da OUTB) |
| 011 | 0B00 | SUB ACC, 0 |  | ; NOP |
| 012 | 0500 | OUTB 0 |  | ; Finaliza o comando (“E” – Enable vai de High para Low) |
| 013 | 0002 | LDI ACC, 2 |  | ; Carrega ACC com 2 (Terceiro Índice de retorno = 2) |
| 014 | 0A0F | STW RAM[0:0:F], ACC |  | ; Salva o índice de retorno na RAM |
| 015 | 300x | ~~LDI RD, Delay\_ms[8:11]~~ |  | ; Carrega o endereço alto da Subrotina em RD |
| 016 | 0C64 | JPI Delay\_ms |  | ; Chama a Subrotina Delay\_ms |
| **RET\_Delay\_ms\_02:** |  |  |  |  |
| ; Envia Comando 28h em D7:D4 | | | | |
| 017 | 0402 | OUTA 2 |  | **; Send Higher Nibble** |
| 018 | 0502 | OUTB 2 |  | ; Prepara o sinal “E” - OUTB[1] = E = 1 (bit 1 da OUTB) |
| 019 | 0B00 | SUB ACC, 0 |  | ; NOP |
| 01A | 0500 | OUTB 0 |  | ; Finaliza o comando (“E” – Enable vai de High para Low) |
| 01B | 0003 | LDI ACC, 3 |  | ; Carrega ACC com 3 (Quarto Índice de retorno = 3) |
| 01C | 0A0F | STW RAM[0:0:F], ACC |  | ; Salva o índice de retorno na RAM |
| 01D | 300x | ~~LDI RD, Delay\_ms[8:11]~~ |  | ; Carrega o endereço alto da Subrotina em RD |
| 01E | 0C64 | JPI Delay\_ms |  | ; Chama a Subrotina Delay\_ms |
| **RET\_Delay\_ms\_03:** |  |  |  |  |
| 01F | 0408 | OUTA 8 |  | **; Send Lower Nibble** |
| 020 | 0502 | OUTB 2 |  | ; Prepara o sinal “E” - OUTB[1] = E = 1 (bit 1 da OUTB) |
| 021 | 0B00 | SUB ACC, 0 |  | ; NOP |
| 022 | 0500 | OUTB 0 |  | ; Finaliza o comando (“E” – Enable vai de High para Low) |
| 023 | 0004 | LDI ACC, 4 |  | ; Carrega ACC com 4 (Quinto Índice de retorno = 4) |
| 024 | 0A0F | STW RAM[0:0:F], ACC |  | ; Salva o índice de retorno na RAM |
| 025 | 300x | ~~LDI RD, Delay\_ms[8:11]~~ |  | ; Carrega o endereço alto da Subrotina em RD |
| 026 | 0C64 | JPI Delay\_ms |  | ; Chama a Subrotina Delay\_ms |
| **RET\_Delay\_ms\_04:** |  |  |  |  |
| ; Envia Comando 0Eh em D7:D4 | | | | |
| 027 | 0400 | OUTA 0 |  | **; Send Higher Nibble** |
| 028 | 0502 | OUTB 2 |  | ; Prepara o sinal “E” - OUTB[1] = E = 1 (bit 1 da OUTB) |
| 029 | 0B00 | SUB ACC, 0 |  | ; NOP |
| 02A | 0500 | OUTB 0 |  | ; Finaliza o comando (“E” – Enable vai de High para Low) |
| 02B | 0005 | LDI ACC, 5 |  | ; Carrega ACC com 5 (Sexto Índice de retorno = 5) |
| 02C | 0A0F | STW RAM[0:0:F], ACC |  | ; Salva o índice de retorno na RAM |
| 02D | 300x | ~~LDI RD, Delay\_ms[8:11]~~ |  | ; Carrega o endereço alto da Subrotina em RD |
| 02E | 0C64 | JPI Delay\_ms |  | ; Chama a Subrotina Delay\_ms |
| **RET\_Delay\_ms\_05:** |  |  |  |  |
| 02F | 040E | OUTA Eh |  | **; Send Lower Nibble** |
| 030 | 0502 | OUTB 2 |  | ; Prepara o sinal “E” - OUTB[1] = E = 1 (bit 1 da OUTB) |
| 031 | 0B00 | SUB ACC, 0 |  | ; NOP |
| 032 | 0500 | OUTB 0 |  | ; Finaliza o comando (“E” – Enable vai de High para Low) |
| 033 | 0006 | LDI ACC, 6 |  | ; Carrega ACC com 6 (Sétimo Índice de retorno = 6) |
| 034 | 0A0F | STW RAM[0:0:F], ACC |  | ; Salva o índice de retorno na RAM |
| 035 | 300x | ~~LDI RD, Delay\_ms[8:11]~~ |  | ; Carrega o endereço alto da Subrotina em RD |
| 036 | 0C64 | JPI Delay\_ms |  | ; Chama a Subrotina Delay\_ms |
| **RET\_Delay\_ms\_06:** |  |  |  |  |
| ; Envia Comando 06 em D7:D4 | | | | |
| 037 | 0400 | OUTA 0 |  | **; Send Higher Nibble** |
| 038 | 0502 | OUTB 2 |  | ; Prepara o sinal “E” - OUTB[1] = E = 1 (bit 1 da OUTB) |
| 039 | 0B00 | SUB ACC, 0 |  | ; NOP |
| 03A | 0500 | OUTB 0 |  | ; Finaliza o comando (“E” – Enable vai de High para Low) |
| 03B | 0007 | LDI ACC, 7 |  | ; Carrega ACC com 7 (Oitavo Índice de retorno = 7) |
| 03C | 0A0F | STW RAM[0:0:F], ACC |  | ; Salva o índice de retorno na RAM |
| 03D | 300x | ~~LDI RD, Delay\_ms[8:11]~~ |  | ; Carrega o endereço alto da Subrotina em RD |
| 03E | 0C64 | JPI Delay\_ms |  | ; Chama a Subrotina Delay\_ms |
| **RET\_Delay\_ms\_07:** |  |  |  |  |
| 03F | 0406 | OUTA 6 |  | **; Send Lower Nibble** |
| 040 | 0502 | OUTB 2 |  | ; Prepara o sinal “E” - OUTB[1] = E = 1 (bit 1 da OUTB) |
| 041 | 0B00 | SUB ACC, 0 |  | ; NOP |
| 042 | 0500 | OUTB 0 |  | ; Finaliza o comando (“E” – Enable vai de High para Low) |
| 043 | 0008 | LDI ACC, 8 |  | ; Carrega ACC com 8 (Nono Índice de retorno = 8) |
| 044 | 0A0F | STW RAM[0:0:F], ACC |  | ; Salva o índice de retorno na RAM |
| 045 | 300x | ~~LDI RD, Delay\_ms[8:11]~~ |  | ; Carrega o endereço alto da Subrotina em RD |
| 046 | 0C64 | JPI Delay\_ms |  | ; Chama a Subrotina Delay\_ms |
| **RET\_Delay\_ms\_08:** |  |  |  |  |
| ; Envia Dado - Letra M em D7:D4 - M = 0100 1101 = 4Dh | | | | |
| 047 | 0404 | OUTA 4 |  | **; Send Higher Nibble** |
| 048 | 0503 | OUTB 3 |  | ; Prepara o sinal E=1 (bit 1 da OUTB) RS=1 (bit 0 da OUTB) |
| 049 | 0B00 | SUB ACC, 0 |  | ; NOP |
| 04A | 0501 | OUTB 1 |  | ; Desliga o terminal “E” e mantém RS=1 (finaliza o cmd) |
| 04B | 0009 | LDI ACC, 9 |  | ; Carrega ACC com 9 (Décimmo Índice de retorno = 9) |
| 04C | 0A0F | STW RAM[0:0:F], ACC |  | ; Salva o índice de retorno na RAM |
| 04D | 300x | ~~LDI RD, Delay\_ms[8:11]~~ |  | ; Carrega o endereço alto da Subrotina em RD |
| 04E | 0C64 | JPI Delay\_ms |  | ; Chama a Subrotina Delay\_ms |
| **RET\_Delay\_ms\_09:** |  |  |  |  |
| 04F | 040D | OUTA Dh |  | **; Send Lower Nibble** |
| 050 | 0503 | OUTB 3 |  | ; Prepara o sinal E=1 (bit 1 da OUTB) RS=1 (bit 0 da OUTB) |
| 051 | 0B00 | SUB ACC, 0 |  | ; NOP |
| 052 | 0501 | OUTB 1 |  | ; Desliga o terminal “E” e mantém RS=1 (finaliza o cmd) |
| 053 | 000A | LDI ACC, Ah |  | ; Carrega ACC com Ah (11º Índice de retorno = Ah) |
| 054 | 0A0F | STW RAM[0:0:F], ACC |  | ; Salva o índice de retorno na RAM |
| 055 | 300x | ~~LDI RD, Delay\_ms[8:11]~~ |  | ; Carrega o endereço alto da Subrotina em RD |
| 056 | 0C64 | JPI Delay\_ms |  | ; Chama a Subrotina Delay\_ms |
| **RET\_Delay\_ms\_0A:** |  |  |  |  |
| ; Envia Dado - Letra i em D7:D4 - i = 0110 1001 = 69h | | | | |
| 057 | 0406 | OUTA 6 |  | **; Send Higher Nibble** |
| 058 | 0503 | OUTB 3 |  | ; Prepara o sinal E=1 (bit1 da OUTB), RS=1 (bit0 da OUTB) |
| 059 | 0B00 | SUB ACC, 0 |  | ; NOP |
| 05A | 0501 | OUTB 1 |  | ; Desliga o terminal “E” e mantém RS=1 (finaliza o cmd) |
| 05B | 000B | LDI ACC, Bh |  | ; Carrega ACC com Bh (12º Índice de retorno = Bh) |
| 05C | 0A0F | STW RAM[0:0:F], ACC |  | ; Salva o índice de retorno na RAM |
| 05D | 300x | ~~LDI RD, Delay\_ms[8:11]~~ |  | ; Carrega o endereço alto da Subrotina em RD |
| 05E | 0C64 | JPI Delay\_ms |  | ; Chama a Subrotina Delay\_ms |
| **RET\_Delay\_ms\_0B:** |  |  |  |  |
| 05F | 0409 | OUTA 9 |  | **; Send Lower Nibble** |
| 060 | 0503 | OUTB 3 |  | ; Prepara o sinal E=1 (bit1 da OUTB), RS=1 (bit0 da OUTB) |
| 061 | 0B00 | SUB ACC, 0 |  | ; NOP |
| 062 | 0501 | OUTB 1 |  | ; Desliga o terminal “E” e mantém RS=1 (finaliza o cmd) |
| 063 | 000C | LDI ACC, Ch |  | ; Carrega ACC com Ch (13º Índice de retorno = Ch) |
| 064 | 0A0F | STW RAM[0:0:F], ACC |  | ; Salva o índice de retorno na RAM |
| 065 | 300x | ~~LDI RD, Delay\_ms[8:11]~~ |  | ; Carrega o endereço alto da Subrotina em RD |
| 066 |  | JPI Delay\_ms |  | ; Chama a Subrotina Delay\_ms |
| **RET\_Delay\_ms\_0C:** |  |  |  |  |
| ; Envia Dado - Letra k em D7:D4 - k = 0110 1011 = 6Bh | | | | |
| 067 | 0406 | OUTA 6 |  | **; Send Higher Nibble** |
| 068 | 0503 | OUTB 3 |  |  |
| 069 | 0B00 | SUB ACC, 0 |  |  |
| 06A | 0501 | OUTB 1 |  |  |
| 06B | 000D | LDI ACC, Dh |  |  |
| 06C | 0A0F | STW RAM[0:0:F], ACC |  |  |
| 06D | 300x | ~~LDI RD, Delay\_ms[8:11]~~ |  |  |
| 06E |  | JPI Delay\_ms |  |  |
| **RET\_Delay\_ms\_0D:** |  |  |  |  |
| 06F | 040B | OUTA Bh |  | **; Send Lower Nibble** |
| 070 | 0503 | OUTB 3 |  |  |
| 071 | 0B00 | SUB ACC, 0 |  |  |
| 072 | 0501 | OUTB 1 |  |  |
| 073 | 000E | LDI ACC, Eh |  |  |
| 074 | 0A0F | STW RAM[0:0:F], ACC |  |  |
| 075 | 300x | ~~LDI RD, Delay\_ms[8:11]~~ |  |  |
| 076 |  | JPI Delay\_ms |  |  |
| **RET\_Delay\_ms\_0E:** |  |  |  |  |
| ; Envia Dado - Letra r em D7:D4 - r = 0111 0010 = 72h | | | | |
| 077 | 0407 | OUTA 7 |  | **; Send Higher Nibble** |
| 078 | 0503 | OUTB 3 |  |  |
| 079 | 0B00 | SUB ACC, 0 |  |  |
| 07A | 0501 | OUTB 1 |  |  |
| 07B | 000F | LDI ACC, Fh |  |  |
| 07C | 0A0F | STW RAM[0:0:F], ACC |  |  |
| 07D | 300x | ~~LDI RD, Delay\_ms[8:11]~~ |  |  |
| 07E |  | JPI Delay\_ms |  |  |
| **RET\_Delay\_ms\_0F:** |  |  |  |  |
| 07F | 0402 | OUTA 2 |  | **; Send Lower Nibble** |
| 080 | 0503 | OUTB 3 |  |  |
| 081 | 0B00 | SUB ACC, 0 |  |  |
| 082 | 0501 | OUTB 1 |  |  |
| 083 | 0001 | LDI ACC, 1 |  |  |
| 084 | 0A0E | STW RAM[0:0:E], ACC |  | ; Salva o índice de retorno na RAM |
| 085 | 300x | ~~LDI RD, Delay\_ms[8:11]~~ |  |  |
| 086 |  | JPI Delay\_ms |  |  |
| **RET\_Delay\_ms\_10:** |  |  |  |  |
| ; Envia Dado - Letra o em D7:D4 - o = 0110 1111 = 6Fh | | | | |
| 087 |  |  |  |  |
| 088 |  |  |  |  |
| 089 |  |  |  |  |
| 08A |  |  |  |  |
| 08B |  |  |  |  |
| 08C |  |  |  |  |
| 08D |  |  |  |  |
| 08E |  |  |  |  |
| **RET\_Delay\_ms\_11:** |  |  |  |  |
| 08F |  |  |  |  |
| 090 |  |  |  |  |
| 091 |  |  |  |  |
| 092 |  |  |  |  |
| 093 |  |  |  |  |
| 094 |  |  |  |  |
| 095 |  |  |  |  |
| 096 |  |  |  |  |
| **RET\_Delay\_ms\_12:** |  |  |  |  |
| ; Envia Dado - Letra E em D7:D4 - E = 0100 0101 = 45h | | | | |
| 097 |  |  |  |  |
| 098 |  |  |  |  |
| 099 |  |  |  |  |
| 09A |  |  |  |  |
| 09B |  |  |  |  |
| 09C |  |  |  |  |
| 09D |  |  |  |  |
| 09E |  |  |  |  |
| 09F |  |  |  |  |
| **RET\_Delay\_ms\_13:** |  |  |  |  |
| 0A0 |  |  |  |  |
| 0A1 |  |  |  |  |
| 0A2 |  |  |  |  |
| 0A3 |  |  |  |  |
| 0A4 |  |  |  |  |
| 0A5 |  |  |  |  |
| 0A6 |  |  |  |  |
| 0A7 |  |  |  |  |
| **RET\_Delay\_ms\_14:** |  |  |  |  |
| ; Envia Dado - Letra u em D7:D4 - u = 0111 0101 = 75h | | | | |
| 0A8 |  |  |  |  |
| 0A9 |  |  |  |  |
| 0AA |  |  |  |  |
| 0AB |  |  |  |  |
| 0AC |  |  |  |  |
| 0AD |  |  |  |  |
| 0AE |  |  |  |  |
| 0AF |  |  |  |  |
| **RET\_Delay\_ms\_15:** |  |  |  |  |
| 0B0 |  |  |  |  |
| 0B1 |  |  |  |  |
| 0B2 |  |  |  |  |
| 0B3 |  |  |  |  |
| 0B4 |  |  |  |  |
| 0B5 |  |  |  |  |
| 0B6 |  |  |  |  |
| 0B7 |  |  |  |  |
| **RET\_Delay\_ms\_16:** |  |  |  |  |
| ; Envia Dado - Letra l em D7:D4 - l = 0110 1100 = 6Ch | | | | |
| 0B8 |  |  |  |  |
| 0B9 |  |  |  |  |
| 0BA |  |  |  |  |
| 0BB |  |  |  |  |
| 0BC |  |  |  |  |
| 0BD |  |  |  |  |
| 0BE |  |  |  |  |
| 0BF |  |  |  |  |
| 0C0 |  |  |  |  |
| **RET\_Delay\_ms\_17:** |  |  |  |  |
| 0C1 |  |  |  |  |
| 0C2 |  |  |  |  |
| 0C3 |  |  |  |  |
| 0C4 |  |  |  |  |
| 0C5 |  |  |  |  |
| 0C6 |  |  |  |  |
| 0C7 |  |  |  |  |
| 0C8 |  |  |  |  |
| **RET\_Delay\_ms\_18:** |  |  |  |  |
| ; Envia Dado - Letra e em D7:D4 - e = 0110 0101 = 65h | | | | |
| 0C9 |  |  |  |  |
| 0CA |  |  |  |  |
| 0CB |  |  |  |  |
| 0CC |  |  |  |  |
| 0CD |  |  |  |  |
| 0CE |  |  |  |  |
| 0CF |  |  |  |  |
| 0D0 |  |  |  |  |
| **RET\_Delay\_ms\_19:** |  |  |  |  |
| 0D1 |  |  |  |  |
| 0D2 |  |  |  |  |
| 0D3 |  |  |  |  |
| 0D4 |  |  |  |  |
| 0D5 |  |  |  |  |
| 0D6 |  |  |  |  |
| 0D7 |  |  |  |  |
| 0D8 |  |  |  |  |
| **RET\_Delay\_ms\_1A:** |  |  |  |  |
| ; Envia Dado - Letra r em D7:D4 - r = 0111 0010 = 72h | | | | |
| 0D9 |  |  |  |  |
| 0DA |  |  |  |  |
| 0DB |  |  |  |  |
| 0DC |  |  |  |  |
| 0DD |  |  |  |  |
| 0DE |  |  |  |  |
| 0DF |  |  |  |  |
| 0E0 |  |  |  |  |
| **RET\_Delay\_ms\_1B:** |  |  |  |  |
| 0E1 |  |  |  |  |
| 0E2 |  |  |  |  |
| 0E3 |  |  |  |  |
| 0E4 |  |  |  |  |
| 0E5 |  |  |  |  |
| 0E6 |  |  |  |  |
| 0E7 |  |  |  |  |
| 0E8 |  |  |  |  |
| **RET\_Delay\_ms\_1C:** |  |  |  |  |
| ; Envia Comando C0h em D7:D4 (Posiciona o cursor no início da segunda linha do LCD) | | | | |
| 0E9 | 040C | OUTA Ch |  | **; Send Higher Nibble** |
| 0EA | 0502 | OUTB 2 |  | ; Prepara o sinal “E” - OUTB[1] = E = 1 (bit 1 da OUTB) |
| 0EB | 0B00 | SUB ACC, 0 |  | ; NOP |
| 0EC | 0500 | OUTB 0 |  |  |
| 0ED | 000x | LDI ACC, 7 |  |  |
| 0EE | 0A0F | STW RAM[0:0:F], ACC |  |  |
| 0EF | 300x | ~~LDI RD, Delay\_ms[8:11]~~ |  |  |
| 0F0 | 0C64 | JPI Delay\_ms |  |  |
| **RET\_Delay\_ms\_1D:** |  |  |  |  |
| 0F1 | 0400 | OUTA 0 |  | **; Send Lower Nibble** |
| 0F2 | 0502 | OUTB 2 |  |  |
| 0F3 | 0B00 | SUB ACC, 0 |  |  |
| 0F4 | 0500 | OUTB 0 |  |  |
| 0F5 | 000x | LDI ACC, 7 |  |  |
| 0F6 | 0A0F | STW RAM[0:0:F], ACC |  |  |
| 0F7 | 300x | ~~LDI RD, Delay\_ms[8:11]~~ |  |  |
| 0F8 | 0C64 | JPI Delay\_ms |  |  |
| **Delay\_ms:** |  |  |  |  |
| // Subrotina de Delay  // RAM[0:0:0] = Contador0  // RAM[0:0:1] = Contador1  // RAM[0:0:2] = Ajuste do Número total de ciclos em Delay2 e Delay3 (deve ser inicializado antes de chamar a subrotina)  // Total de chamadas possíveis: 32 (0 a 31)  // RAM[0:0:F] = Índice de retorno para as 16 primeiras chamadas da subrotina (chamadas 0 a 15)  // RAM[0:0:E] = Índice de retorno para as 16 próximas chamadas da subrotina (chamadas 16 a 31) | | | | |
| 0F9 | 0000 | LDI ACC, 0 |  | // Prepara para inicializar os contadores |
| 0FA | 0A00 | STW RAM[0:0:0], ACC |  | // Inicializa Contador0 em 0 |
| 0FB | 0A01 | STW RAM[0:0:1], ACC |  | // Inicializa Contador1 em 0 |
| **Delay1\_ms:** |  |  |  |  |
| 0FC | 080F | CMP ACC, Fh |  | // Fim do Delay1\_ms ? |
| 0FD | 0E6F | JPZ Delay2\_ms |  | // Sim |
| 0FE | 0F01 | ADD ACC, 1 |  | // Não, incrementa ACC |
| 0FF | 0B00 | SUB ACC, 0 |  | // NOP |
| 100 | 0B00 | SUB ACC, 0 |  | // NOP |
| 101 | 0B00 | SUB ACC, 0 |  | // NOP |
| 102 | 0B00 | SUB ACC, 0 |  | // NOP |
| 103 | 0C67 | JPI Delay1\_ms |  | // Loop |
| **Delay2\_ms:** |  |  |  |  |
| 104 | 0200 | LDW ACC, RAM[0:0:0] |  | // Carrega o valor do Contador0 em ACC |
| 105 | 3802 | CMP ACC, RAM[0:0:2] |  | // Contador0 é igual ao número de ciclos ajustado ? |
| 106 | 0E7A | JPZ Delay3\_ms |  | // Sim |
| 107 | 0F01 | ADD ACC, 1 |  | // Não, incrementa ACC |
| 108 | 0A00 | STW RAM[0:0:0], ACC |  | // Atualiza o contador0 |
| 109 | 0B00 | SUB ACC, 0 |  | // NOP |
| 10A | 0B00 | SUB ACC, 0 |  | // NOP |
| 10B | 0B00 | SUB ACC, 0 |  | // NOP |
| 10C | 0B00 | SUB ACC, 0 |  | // NOP |
| 10D | 0000 | LDI ACC, 0 |  | // Reinicializa o ACC |
| 10E | 0C67 | JPI Delay1\_ms |  |  |
| **Delay3\_ms:** |  |  |  |  |
| 10F | 0201 | LDW ACC, RAM[0:0:1] |  | // Carrega o valor do Contador1 em ACC |
| 110 | 3802 | CMP ACC, RAM[0:0:2] |  | // Contador1 é Igual ao número de ciclos ajustado ? |
| 111 | 0E86 | JPZ DelayEnd\_ms |  | // Sim |
| 112 | 0F01 | ADD ACC, 1 |  | // Não, incrementa ACC |
| 113 | 0A01 | STW RAM[0:0:1], ACC |  | // Atualiza o contador1 |
| 114 | 0B00 | SUB ACC, 0 |  | // NOP |
| 115 | 0B00 | SUB ACC, 0 |  | // NOP |
| 116 | 0B00 | SUB ACC, 0 |  | // NOP |
| 117 | 0B00 | SUB ACC, 0 |  | // NOP |
| 118 | 0000 | LDI ACC, 0 |  | // Reinicializa ACC |
| 119 | 0A00 | STW RAM[0:0:0], ACC |  | // Reinicializa o Contador0 |
| 11A | 0C6F | JPI Delay2\_ms |  |  |
| **DelayEnd\_ms:** |  |  |  |  |
| 11B | 020F | LDW ACC, RAM[0:0:F] |  | // Recupera endereço de retorno |
| 11C | 080F | CMP ACC, Fh |  | ; Verifica se as 16 primeiras chamadas foram feitas |
| 11D | 0E9x | JPZ DelayEnd\_ms\_10 |  | ; Sim, vai para as próximas 16 chamadas |
| 11E | 020F | LDW ACC, RAM[0:0:F] |  | // Não, recupera endereço de retorno |
| 11F | 0E9D | JPZ Delay\_ms\_RET\_00 |  | // Se Z = 1, índice de retorno = 0, vai para o ponto RET-0 |
| 120 | 0801 | CMP ACC, 1 |  | // ACC=1 ? – Segundo Índice de retorno |
| 121 | 0E9F | JPZ Delay\_ms\_RET\_01 |  | // Se Z = 1, índice de retorno = 1, vai para o ponto RET-1 |
| 122 | 0802 | CMP ACC, 2 |  | // ACC=2 ? – Terceiro Índice de retorno |
| 123 | 0EA1 | JPZ Delay\_ms\_RET\_02 |  | // Se Z = 1, índice de retorno = 2, vai para o ponto RET-2 |
| 124 | 0803 | CMP ACC, 3 |  |  |
| 125 | 0EA3 | JPZ Delay\_ms\_RET\_03 |  |  |
| 126 | 0804 | CMP ACC, 4 |  |  |
| 127 | 0EA5 | JPZ Delay\_ms\_RET\_04 |  |  |
| 128 | 0805 | CMP ACC, 5 |  |  |
| 129 | 0EA7 | JPZ Delay\_ms\_RET\_05 |  |  |
| 12A | 0806 | CMP ACC, 6 |  |  |
| 12B | 0EA9 | JPZ Delay\_ms\_RET\_06 |  |  |
| 12C | 0807 | CMP ACC, 7 |  |  |
| 12D | 0EAB | JPZ Delay\_ms\_RET\_07 |  |  |
| 12E | 0808 | CMP ACC, 8 |  |  |
| 12F | 0EAD | JPZ Delay\_ms\_RET\_08 |  |  |
| 130 | 0809 | CMP ACC, 9 |  |  |
| 131 | 0EAF | JPZ Delay\_ms\_RET\_09 |  |  |
| 132 | 080A | CMP ACC, Ah |  |  |
| 133 | 0EB1 | JPZ Delay\_ms\_RET\_0A |  |  |
| 134 |  | CMP ACC, Bh |  |  |
| 135 |  | JPZ Delay\_ms\_RET\_0B |  |  |
| 136 |  | CMP ACC, Ch |  |  |
| 137 |  | JPZ Delay\_ms\_RET\_0C |  |  |
| 138 |  | CMP ACC, Dh |  |  |
| 139 |  | JPZ Delay\_ms\_RET\_0D |  |  |
| 13A |  | CMP ACC, Eh |  |  |
| 13B |  | JPZ Delay\_ms\_RET\_0E |  |  |
| 13C |  | CMP ACC, Fh |  |  |
| 13D |  | JPZ Delay\_ms\_RET\_0F |  |  |
| **DelayEnd\_ms\_10:** |  |  |  | ; Próximas 16 chamadas da subrotina => índice = RAM[E] |
| 13E |  | LDW ACC, RAM[0:0:E] |  | ; Recupera endereço de retorno |
| 13F |  | JPZ Delay\_ms\_RET\_10 |  | // Se Z = 1, índice de retorno = 0, vai para o ponto RET-10 |
| 140 |  | CMP ACC, 1 |  |  |
| 141 |  | JPZ Delay\_ms\_RET\_11 |  |  |
| 142 |  | CMP ACC, 2 |  |  |
| 143 |  | JPZ Delay\_ms\_RET\_12 |  |  |
| 144 |  | CMP ACC, 3 |  |  |
| 145 |  | JPZ Delay\_ms\_RET\_13 |  |  |
| 146 |  | CMP ACC, 4 |  |  |
| 147 |  | JPZ Delay\_ms\_RET\_14 |  |  |
| 148 |  | CMP ACC, 5 |  |  |
| 149 |  | JPZ Delay\_ms\_RET\_15 |  |  |
| 14A |  | CMP ACC, 6 |  |  |
| 14B |  | JPZ Delay\_ms\_RET\_16 |  |  |
| 14C |  | CMP ACC, 7 |  |  |
| 14D |  | JPZ Delay\_ms\_RET\_17 |  |  |
| 14E |  | CMP ACC, 8 |  |  |
| 14F |  | JPZ Delay\_ms\_RET\_18 |  |  |
| 150 |  | CMP ACC, 9 |  |  |
| 151 |  | JPZ Delay\_ms\_RET\_19 |  |  |
| 152 |  | CMP ACC, Ah |  |  |
| 153 |  | JPZ Delay\_ms\_RET\_1A |  |  |
| 154 |  | CMP ACC, Bh |  |  |
| 155 |  | JPZ Delay\_ms\_RET\_1B |  |  |
| 156 |  | CMP ACC, Ch |  |  |
| 157 |  | JPZ Delay\_ms\_RET\_1C |  |  |
| 158 |  | CMP ACC, Dh |  |  |
| 159 |  | JPZ Delay\_ms\_RET\_1D |  |  |
| 15A |  | CMP ACC, Eh |  |  |
| 15B |  | JPZ Delay\_ms\_RET\_1E |  |  |
| 15C |  | CMP ACC, Fh |  |  |
| 15D |  | JPZ Delay\_ms\_RET\_1F |  |  |
| **HALT:** |  |  |  |  |
| 15E | 0C9C | JPI HALT |  | // Endereço de retorno desconhecido, para a execução |
| **Delay\_ms\_RET\_00:** |  |  |  |  |
| 15F | 300x | ~~LDI RD, #RET-0[8:11]~~ |  |  |
| 160 | 0C06 | JPI RET\_Delay\_ms\_00 |  |  |
| **Delay\_ms\_RET\_01:** |  |  |  |  |
| 161 | 300x | ~~LDI RD, #RET-1[8:11]~~ |  |  |
| 162 | 0C0F | JPI RET\_Delay\_ms\_01 |  |  |
| **Delay\_ms\_RET\_02:** |  |  |  |  |
| 163 | 300x | ~~LDI RD, #RET-2[8:11]~~ |  |  |
| 164 | 0C17 | JPI RET\_Delay\_ms\_02 |  |  |
| **Delay\_ms\_RET\_03:** |  |  |  |  |
| 165 | 300x | ~~LDI RD, #RET-2[8:11]~~ |  |  |
| 166 | 0C1F | JPI RET\_Delay\_ms\_03 |  |  |
| **Delay\_ms\_RET\_04:** |  |  |  |  |
| 167 | 300x | ~~LDI RD, #RET-2[8:11]~~ |  |  |
| 168 | 0C27 | JPI RET\_Delay\_ms\_04 |  |  |
| **Delay\_ms\_RET\_05:** |  |  |  |  |
| 169 | 300x | ~~LDI RD, #RET-2[8:11]~~ |  |  |
| 16A | 0C2F | JPI RET\_Delay\_ms\_05 |  |  |
| **Delay\_ms\_RET\_06:** |  |  |  |  |
| 16B | 300x | ~~LDI RD, #RET-2[8:11]~~ |  |  |
| 16C | 0C37 | JPI RET\_Delay\_ms\_06 |  |  |
| **Delay\_ms\_RET\_07:** |  |  |  |  |
| 16D | 300x | ~~LDI RD, #RET-2[8:11]~~ |  |  |
| 16E | 0C3F | JPI RET\_Delay\_ms\_07 |  |  |
| **Delay\_ms\_RET\_08:** |  |  |  |  |
| 16F | 300x | ~~LDI RD, #RET-2[8:11]~~ |  |  |
| 170 | 0C47 | JPI RET\_Delay\_ms\_08 |  |  |
| **Delay\_ms\_RET\_09:** |  |  |  |  |
| 171 | 300x | ~~LDI RD, #RET-2[8:11]~~ |  |  |
| 172 | 0C4F | JPI RET\_Delay\_ms\_09 |  |  |
| **Delay\_ms\_RET\_0A:** |  |  |  |  |
| 173 | 300x | ~~LDI RD, #RET-2[8:11]~~ |  |  |
| 174 | 0C57 | JPI RET\_Delay\_ms\_0A |  |  |
| **Delay\_ms\_RET\_0B:** |  |  |  |  |
| 175 | 300x | ~~LDI RD, #RET-2[8:11]~~ |  |  |
| 176 | 0C57 | JPI RET\_Delay\_ms\_0B |  |  |
| **Delay\_ms\_RET\_0C:** |  |  |  |  |
| 177 | 300x | ~~LDI RD, #RET-2[8:11]~~ |  |  |
| 178 | 0C57 | JPI RET\_Delay\_ms\_0C |  |  |
| **Delay\_ms\_RET\_0D:** |  |  |  |  |
| 179 | 300x | ~~LDI RD, #RET-2[8:11]~~ |  |  |
| 17A | 0C57 | JPI RET\_Delay\_ms\_0D |  |  |
| **Delay\_ms\_RET\_0E:** |  |  |  |  |
| 17B | 300x | ~~LDI RD, #RET-2[8:11]~~ |  |  |
| 17C | 0C57 | JPI RET\_Delay\_ms\_0E |  |  |
| **Delay\_ms\_RET\_0F:** |  |  |  |  |
| 17D | 300x | ~~LDI RD, #RET-2[8:11]~~ |  |  |
| 17E | 0C57 | JPI RET\_Delay\_ms\_0F |  |  |
| **Delay\_ms\_RET\_10:** |  |  |  |  |
| 17F | 300x | ~~LDI RD, #RET-2[8:11]~~ |  |  |
| 180 | 0C57 | JPI RET\_Delay\_ms\_10 |  |  |
| **Delay\_ms\_RET\_11:** |  |  |  |  |
| 181 |  | ~~LDI RD, #RET-2[8:11]~~ |  |  |
| 182 |  | JPI RET\_Delay\_ms\_11 |  |  |
| **Delay\_ms\_RET\_12:** |  |  |  |  |
| 183 |  | ~~LDI RD, #RET-2[8:11]~~ |  |  |
| 184 |  | JPI RET\_Delay\_ms\_12 |  |  |
| **Delay\_ms\_RET\_13:** |  |  |  |  |
| 185 |  | ~~LDI RD, #RET-2[8:11]~~ |  |  |
| 186 |  | JPI RET\_Delay\_ms\_13 |  |  |
| **Delay\_ms\_RET\_14:** |  |  |  |  |
| 187 |  | ~~LDI RD, #RET-2[8:11]~~ |  |  |
| 188 |  | JPI RET\_Delay\_ms\_14 |  |  |
| **Delay\_ms\_RET\_15:** |  |  |  |  |
| 189 |  | ~~LDI RD, #RET-2[8:11]~~ |  |  |
| 18A |  | JPI RET\_Delay\_ms\_15 |  |  |
| **Delay\_ms\_RET\_16:** |  |  |  |  |
| 18B |  | ~~LDI RD, #RET-2[8:11]~~ |  |  |
| 18C |  | JPI RET\_Delay\_ms\_16 |  |  |
| **Delay\_ms\_RET\_17:** |  |  |  |  |
| 18D |  | ~~LDI RD, #RET-2[8:11]~~ |  |  |
| 18E |  | JPI RET\_Delay\_ms\_17 |  |  |
| **Delay\_ms\_RET\_18:** |  |  |  |  |
| 18F |  | ~~LDI RD, #RET-2[8:11]~~ |  |  |
| 190 |  | JPI RET\_Delay\_ms\_18 |  |  |
| **Delay\_ms\_RET\_19:** |  |  |  |  |
| 191 |  | ~~LDI RD, #RET-2[8:11]~~ |  |  |
| 192 |  | JPI RET\_Delay\_ms\_19 |  |  |
| **Delay\_ms\_RET\_1A:** |  |  |  |  |
| 193 |  | ~~LDI RD, #RET-2[8:11]~~ |  |  |
| 194 |  | JPI RET\_Delay\_ms\_1A |  |  |
| **Delay\_ms\_RET\_1B:** |  |  |  |  |
| 195 |  | ~~LDI RD, #RET-2[8:11]~~ |  |  |
| 196 |  | JPI RET\_Delay\_ms\_1B |  |  |
| **Delay\_ms\_RET\_1C:** |  |  |  |  |
| 197 |  | ~~LDI RD, #RET-2[8:11]~~ |  |  |
| 198 |  | JPI RET\_Delay\_ms\_1C |  |  |
| **Delay\_ms\_RET\_1D:** |  |  |  |  |
| 199 |  | ~~LDI RD, #RET-2[8:11]~~ |  |  |
| 19A |  | JPI RET\_Delay\_ms\_1D |  |  |
| **Delay\_ms\_RET\_1E:** |  |  |  |  |
| 19B |  | ~~LDI RD, #RET-2[8:11]~~ |  |  |
| 19C |  | JPI RET\_Delay\_ms\_1E |  |  |
| **Delay\_ms\_RET\_1F:** |  |  |  |  |
| 19D |  | ~~LDI RD, #RET-2[8:11]~~ |  |  |
| 19E |  | JPI RET\_Delay\_ms\_1F |  |  |

**Delay\_ms:**

// RAM[0:0:0] = Contador0

// RAM[0:0:1] = Contador1

// RAM[0:0:2] = Ajuste do Número total de ciclos em Delay2 (deve ser inicializado antes de chamar a subrotina)

~~// Delay máximo = 50000 ciclos~~

~~// Delay mínimo = 10 ciclos => RAM[0:0:2]=0~~

// Antes de chamar a rotina de atraso, ajustar RD:RB:RA com o endereço de retorno (consome 3 ciclos)

02 OUTA 1 // Somente para teste da rotina de delay - Apagar

03 LDI ACC, 0

04 STW 0:0:0, ACC // Inicializa Contador0 em 0

05 STW 0:0:1, ACC // Inicializa Contador1 em 0

**Delay1\_ms**:

// Número máximo de ciclo: Número Instruções \* 15 repetições (para ACC de 0 a F) + 2 Instruções (quando ACC=Fh)

// 8\*15 + 2 = 122 ciclos

// Número máximo de ciclos=122, ACC=Fh

06 CMP ACC, Fh

07 JPZ Delay2\_ms

08 ADD ACC, 1

09 SUB ACC, 0 // NOP ?

0A SUB ACC, 0 // NOP ?

0B SUB ACC, 0 // NOP ?

0C SUB ACC, 0 // NOP ?

0D JPI Delay1\_ms

**Delay2\_ms**:

// número máximo de ciclos =

// RAM[0:0:2]=0 => 0\*122 ciclos + 3 ciclos = 3 ciclos

// RAM[0:0:2]=1 => 1\*122 ciclos + 11\*1 + 3 = 88 ciclos

// RAM[0:0:2]=2 => 2\*122 ciclos + 11\*2 + 3 = 173 ciclos

// RAM[0:0:2]=4 => 4\*122 ciclos + 11\*4 + 3 = 173 ciclos

// RAM[0:0:2]=8 => 8\*122 ciclos + 11\*8 + 3 = 1067 ciclos

// RAM[0:0:2]=Fh => Fh\*122 ciclos + 11\*Fh + 3 = 1998 ciclos

0E LDW ACC, 0:0:0 // Carrega o valor do Contador0 em ACC

0F CMP ACC, 0:0:2 // Igual o número de ciclos ajustado ?

10 JPZ Delay3\_ms // DelayEnd\_ms

11 ADD ACC, 1 // Incrementa ACC

12 STW 0:0:0, ACC // Atualiza o contador0

13 SUB ACC, 0 // NOP ?

14 SUB ACC, 0 // NOP ?

15 SUB ACC, 0 // NOP ?

16 SUB ACC, 0 // NOP ?

17 LDI ACC, 0 // Reinicializa o ACC

18 JPI Delay1\_ms

**Delay3\_ms:**

// RAM[0:0:2]=Fh => Fh\*1998 ciclos + 11\*Fh + 3 = 30138 ciclos

19 LDW ACC, 0:0:1 // Carrega o valor do Contador1 em ACC

1A CMP ACC, 0:0:2 // Igual o número de ciclos ajustado ?

1B JPZ DelayEnd\_ms

1C ADD ACC, 1 // Incrementa ACC

1D STW 0:0:1, ACC // Atualiza o contador1

1E SUB ACC, 0 // NOP ?

1F SUB ACC, 0 // NOP ?

20 SUB ACC, 0 // NOP ?

21 SUB ACC, 0 // NOP ?

22 LDI ACC, 0 // Limpa o ACC

23 STW 0:0:0, ACC // Reinicializa o Contador0

24 JPI Delay2\_ms

**DelayEnd\_ms:**

25 OUTA 0 // Somente para teste da rotina de delay – Apagar

26 OUTB 0