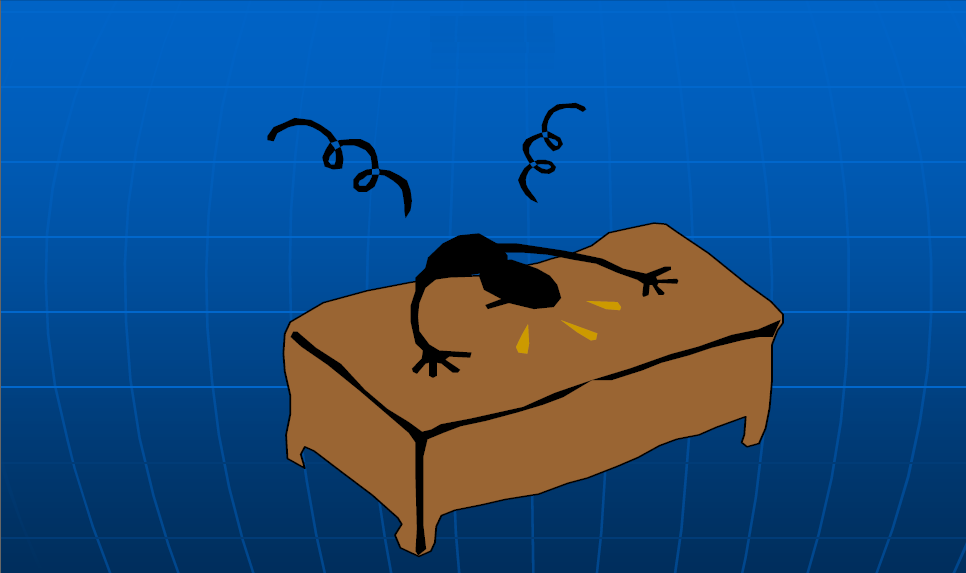


Yhan Christian Souza Silva n°35 3A3

Exercícios 8051 – Professor Silvio



Santos

2013

Exercício 1:

========================================================================

; ================= ETEC Aristóteles Ferreira. ============================

; ======== Yhan Christian Souza Silva =================================

; ===== Programa\_01 exercicio1.asm ver. 0.0 data: 27/04/13. ======================

;========================================================================

;Um painel com três botões para controlar duas esteiras de alimentação de uma máquina

;, dependendo do tipo de peça a ser produzida ligamos manualmente uma das duas esteiras de matéria prima ou as duas juntas

;, além disso, existe um botão único para o desligamento das duas esteiras,

; no painel também existem dois led’s para sinalização de qual esteira está acionada.

MOV P0,#00H ;Limpo P0

MOV P2,#0FFH ;Preparo P2 para leitura

LOOP:

JB P2.0,LOOP ;Verifico o acionamento do botão P2.0

LED\_01: SETB P0.0 ;Coloco em nível lógico 1 P0.0

JB P2.1, LED\_01 ;Verifico o acionamento do botão P2.1

LED\_02: SETB P0.1 ;Coloco em nivel lógico 1 PO.

JB P2.2, LED\_02 ;Verifico o acionamento do botão P2.2

LED\_03: CLR P0.0 ;Limpo P0.0

CLR P0.1 ;Limpo P0.1

MOV P0,#00H ;Limpo o port

LCALL TEMPO ;Chamo subrotina de tempo para atraso SJMP LOOP ;Volto ao Loop

;SUBROTINA TEMPO

TEMPO: MOV R0, #8 ;CARREGA RO COM 8

VOLTA2: MOV R1, #250 ;CARREGA R1 COM 250

VOLTA1: MOV R2, #250 ;CARREGA R2 COM 250

DJNZ R2, $ ;DECREMENTA R2 ATÉ ZERAR

DJNZ R1, VOLTA1 ;DECREMENTA R1 E SALTA ATÉ ZERAR

DJNZ R0, VOLTA2 ;DECREMENTA RO E SALTA ATÉ ZERAR

RET ;RETORNO DE SUBROTINA

END ;FIM DO PROGRAMA

Exercício 2:

========================================================================

; ================= ETEC Aristóteles Ferreira. ========================

; ================ Yhan Christian Souza Silva. ================

; =============== exercicio2.asm data: 04/05/13. ======================

;========================================================================

; Criar um programa que a cada tecla pressionada em P2

; acende o respectivo led em P0, ou seja,

; utilizar uma rotina de varredura de teclado.

;Acionando os botões P2.0 E P2.7 deverá acionar o Buzzer e acender todos os Leds em P0

ORG 0000H ;INICIO DE PROGRAMA

INICIO: MOV P0,#00H ;LIMPA P0

MOV P2,#0FFH ;PREPARA P2 PARA LEITURA

S0: JB P2.0, LIMPA ;VERIFICA O ACIONAMENTO DO BOTAO P2.0 / SE NÃO TIVER VAI PRA ROTINA

SETB P0.0 ;SETA A PORTA P0.0

LIMPA: JNB P2.0, S1 ;VERIFICA O DESACIONAMENTO DO BOTAO P2.0 / SE NÃO TIVER VAI PRA ROTINA

CLR P0.0 ;LIMPA A PORTA P0.0

S1: JB P2.1, LIMPA1 ;VERIFICA O ACIONAMENTO DO BOTAO P2.1 / SE NÃO TIVER VAI PRA ROTINA

SETB P0.1 ;SETA A PORTA P0.1

LIMPA1: JNB P2.1, S2 ;VERIFICA O DESACIONAMENTO DO BOTAO P2.1 / SE NÃO TIVER VAI PRA ROTINA

CLR P0.1 ;LIMPA A PORTA P0.1

S2: JB P2.2, LIMPA2 ;VERIFICA O ACIONAMENTO DO BOTAO P2.2 / SE NÃO TIVER VAI PRA ROTINA

SETB P0.2 ;SETA A PORTA P0.2

LIMPA2: JNB P2.2, S3 ;VERIFICA O DESACIONAMENTO DO BOTAO P2.2 / SE NÃO TIVER VAI PRA ROTINA

CLR P0.2 ;LIMPA A PORTA P0.2

S3: JB P2.3, LIMPA3 ;VERIFICA O ACIONAMENTO DO BOTAO P2.3 / SE NÃO TIVER VAI PRA ROTINA

SETB P0.3 ;SETA A PORTA P0.3

LIMPA3: JNB P2.3, S4 ;VERIFICA O DESACIONAMENTO DO BOTAO P2.3 / SE NÃO TIVER VAI PRA ROTINA

CLR P0.3 ;LIMPA A PORTA P0.3

S4: JB P2.4, LIMPA4 ;VERIFICA O ACIONAMENTO DO BOTAO P2.4 / SE NÃO TIVER VAI PRA ROTINA

SETB P0.4 ;SETA A PORTA P0.4

LIMPA4: JNB P2.4, S5 ;VERIFICA O DESACIONAMENTO DO BOTAO P2.4 / SE NÃO TIVER VAI PRA ROTINA

CLR P0.4 ;LIMPA A PORTA P0.4

S5: JB P2.5, LIMPA5 ;VERIFICA O ACIONAMENTO DO BOTAO P2.5 / SE NÃO TIVER VAI PRA ROTINA

SETB P0.5 ;SETA A PORTA P0.5

LIMPA5: JNB P2.5, S6 ;VERIFICA O DESACIONAMENTO DO BOTAO P2.5 / SE NÃO TIVER VAI PRA ROTINA

CLR P0.5 ;LIMPA A PORTA P0.5

S6: JB P2.6, LIMPA6 ;VERIFICA O ACIONAMENTO DO BOTAO P2.6 / SE NÃO TIVER VAI PRA ROTINA

SETB P0.6 ;SETA A PORTA P0.6

LIMPA6: JNB P2.6, S7 ;VERIFICA O DESACIONAMENTO DO BOTAO P2.6 / SE NÃO TIVER VAI PRA ROTINA

CLR P0.6 ;LIMPA A PORTA P0.6

S7: JB P2.7, LIMPA7 ;VERIFICA O ACIONAMENTO DO BOTAO P2.7 / SE NÃO TIVER VAI PRA ROTINA

SETB P0.7 ; SETA A PORTA P0.7

LIMPA7: JNB P2.7, DUASCHAVES ;VERIFICA O DESACIONAMENTO DO BOTAO P2.7 / SE NÃO TIVER VAI PRA ROTINA

CLR P0.7 ;LIMPA A PORTA P0.7

DUASCHAVES: JB P2.0, DESLIGA ;VERIFICA O ACIONAMENTO DO BOTAO P2.2 / SE NÃO TIVER VAI PRA ROTINA

JB P2.7, DESLIGA ;VERIFICA O ACIONAMENTO DO BOTAO P2.7 / SE NÃO TIVER VAI PRA ROTINA

CLR P3.7 ;LIMPA P3.7

MOV A, #0FFH ;MOVE PARA O ACUMULADOR O DADO 0FFH

MOV P0, A ;MOVE O DADO DO ACUMULADOR PARA P0

SJMP INICIO

DESLIGA: SETB P3.7 ;SETO O BIT DO BUZZER, FAZENDO ELE PARAR DE TOCAR

SJMP INICIO ;PULO PARA INICIO DE PROGRAMA

END ;TERMINA O PROGRAMA

Fluxograma:

Exercício 3:

;========================================================================

; ================= ETEC Aristóteles Ferreira. ==================

; ======== Yhan Christian Souza Silva ===================

; ===== Programa\_01 exercicio3.asm ver. 0.0 data: 28/04/13. ========

; ========================================================================

;Criar um programa para acionar um potenciômetro digital

; com incremento na da tecla P2.0 e decremento com a tecla P2.7

ORG 0000H ;INICIO DO PROGRAMA

MOV P0,#00H ;LIMPA P0

MOV P2,#0FFH ;PREPARA P2 PARA A LEITURA

MOV A,#00H ;LIMPA O ACUMULADOR

LOOP: ;ROTINA DE LOOP

INCREMENTA: JB P2.0, DECREMENTA ;VERIFICA SE P2.0 ESTA ACIONADO / SE NÃO ESTIVER VAI PARA A ROTINA DECREMENTA

VOLTA3: CJNE A,#0FFH,FAZ ;COMPARO O DADO FF COM ACUMULADOR SE NÃO FOR VOU PARA SUBROTINA FAZ

SJMP DECREMENTA ;PULO PARA ROTINA DECREMENTA

FAZ: INC A ;INCREMENTO 1 AO ACC

MOV P0, A ;MOVO O DADO DE ACC PARA P0

LCALL TEMPO ;SUBROTINA DE TEMPO

SJMP INCREMENTA ;PULO PARA ROTINA INCREMENTA

DECREMENTA:JB P2.7, INCREMENTA ;VERIFICA SE P2.7 ESTA ACIONADO / SE NÃO ESTIVER VAI PARA A ROTINA INCREMENTE

VOLTA4: CJNE A,#00H,FAZ2 ;COMPARO O DADO 00 COM ACUMULADOR SE NÃO FOR VOU PARA SUBROTINA FAZ2

SJMP INCREMENTA ;PULO PARA ROTINA INCREMENTA

FAZ2: DEC A ;DECREMENTO 1 AO ACC

MOV P0, A ;MOVO O DADO DE ACC PARA P0

LCALL TEMPO ;SUBROTINA DE TEMPO

SJMP DECREMENTA ;PULO PARA ROTINA DECREMENTA

SJMP LOOP ;VOLTAR PARA LOOP

;SUBROTINA TEMPO

TEMPO: MOV R0, #1 ;CARREGA RO COM 1

VOLTA2: MOV R1, #250 ;CARREGA R1 COM 250

VOLTA1: MOV R2, #250 ;CARREGA R2 COM 250

DJNZ R2, $ ;DECREMENTA R2 ATÉ ZERAR

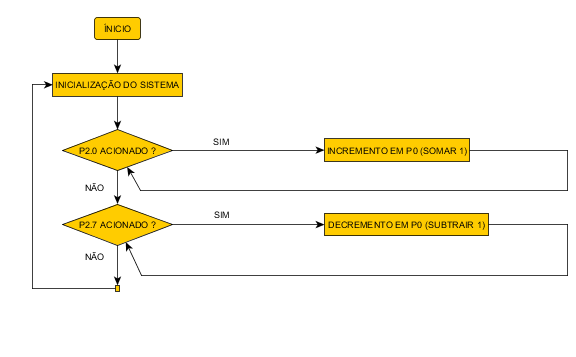
DJNZ R1, VOLTA1 ;DECREMENTA R1 E SALTA ATÉ ZERAR

DJNZ R0, VOLTA2 ;DECREMENTA RO E SALTA ATÉ ZERAR

RET ;RETORNO DE SUBROTINA

END ;FIM DE PROGRAMA

Fluxograma:



Exercício 4:

;========================================================================

; ================== ETEC Aristóteles Ferreira. ==================

; ======== Yhan Christian Souza Silva =================

; ====== Exercicio\_04 alarme.asm ver. 0.0 data: 07/05/13. =======

;========================================================================

ORG 0000 ;Inicio de Programa

MOV P0, #00H ;Limpo P0

MOV P2, #0FFH ;Preparo P2.0 para leitura

SETB P3.7 ;Setar P3.7

ACIONAR00:JB P3.2, ACIONAR00 ;Verificar acionamento de P3.2, pular para Acionar00

JB P2.0, ACIONAR00 ;Verificar acionamento de P2.0, pular para Acionar00

SETB P0.0 ;Setar P0.0

LCALL TEMPOV ;Sub-rotina TempoV

CLR P3.7 ;Limpar P3.7

DESLIGAR: JB P3.2, DESLIGAR ;Verificar acionamento de P3.2, pular para Desligar

SETB P3.7 ;Setar P3.7

SJMP ACIONAR0 ;Pulo para Acionar0

DESLIGA1: MOV P0, #00H ;Movo para P0 o dado 00H

SETB P3.7 ;Setar P3.7

SJMP ACIONAR00 ;Pulo para Acionar00

FECHARP: JB P2.0, ACIONAR1 ;Verificar acionamento de P2.0, pular para Acionar1

SJMP FECHARP ;Pulo para Fecharp

ACIONAR0: CLR P0.0 ;Limpo P0.0

ACIONAR1: JNB P2.0, LED0 ;Verifico se está em nível lógico 0 P2.0, pulo Led0

ACIONAR2: JNB P2.1, LED1 ;Verifico se está em nível lógico 0 P2.1, pulo Led1

ACIONAR3: JNB P2.2, LED2 ;Verifico se está em nível lógico 0 P2.2, pulo Led2

ACIONAR4: JNB P2.3, LED3 ;Verifico se está em nível lógico 0 P2.3, pulo Led3

ACIONAR5: JNB P2.4, LED4 ;Verifico se está em nível lógico 0 P2.4, pulo Led4

ACIONAR6: JNB P2.5, LED5 ;Verifico se está em nível lógico 0 P2.5, pulo Led5

ACIONAR7: JNB P2.6, LED6 ;Verifico se está em nível lógico 0 P2.6, pulo Led6

ACIONAR8: JNB P2.7, LED7 ;Verifico se está em nível lógico 0 P2.7, pulo Led7

SJMP ACIONAR1 ;Pulo para Acionar1

LED0: SETB P0.0 ;Seto P0.0

LCALL TEMPOV ;Sub-rotina de tempo

CLR P3.7 ;Limpo P3.7

JB P3.2, ACIONAR2 ;Verificar acionamento de P3.2, pular para Acionar2

DESLIGA2: SJMP DESLIGA1 ;Pulo para Desliga1

LED1: SETB P0.1 ;Seto P0.1

LCALL TEMPON ;Sub-rotina de tempo

CLR P3.7 ;Limpo P3.7

SJMP ACIONAR3 ;Pulo para Acionar3

LED2: SETB P0.2 ;Seto P0.2

LCALL TEMPON ;Sub-rotina de tempo

CLR P3.7 ;Limpo P3.7

SJMP ACIONAR4 ;Pulo para Acionar4

LED3: SETB P0.3 ;Seto P0.3

LCALL TEMPON ;Sub-rotina de tempo

CLR P3.7 ;Limpo P3.7

SJMP ACIONAR5 ;Pulo para Acionar5

LED4: SETB P0.4 ;Seto P0.4

LCALL TEMPON ;Sub-rotina de tempo

CLR P3.7 ;Limpo P3.7

SJMP ACIONAR6 ;Pulo para Acionar6

LED5: SETB P0.5 ;Seto P0.5

LCALL TEMPON ;Sub-rotina de tempo

CLR P3.7 ;Limpo P3.7

SJMP ACIONAR7 ;Pulo para Acionar7

LED6: SETB P0.6 ;Seto P0.6

LCALL TEMPON ;Sub-rotina de tempo

CLR P3.7 ;Limpo P3.7

SJMP ACIONAR1 ;Pulo para Acionar1

LED7: SETB P0.7 ;Seto P0.7

LCALL TEMPON ;Sub-rotina de tempo

CLR P3.7 ;Limpo P3.7

SJMP ACIONAR2 ;Pulo para Acionar2

;SUBROTINA TEMPO=(250X250X240X2)uS= 30000000uS= 30s.

TEMPOV: JNB P3.2, DESLIGA2 ;Verifico se está em nível lógico 0 P3.2, pulo Desliga2

MOV R0, #240 ;Carrega R0 com 240.

VOLTA2: JNB P3.2, DESLIGA2 ;Verifico se está em nível lógico 0 P3.2, pulo Desliga2

JB P2.0, ACIONAR0 ;Verificar acionamento de P2.0, pular para Acionar0

MOV R1, #250 ;Carrega R1 com 250.

VOLTA1: JNB P3.2, DESLIGA2 ;Verifico se está em nível lógico 0 P3.2, pulo Desliga2

JB P2.0, ACIONAR0 ;Verificar acionamento de P2.0, pular para Acionar0

MOV R2, #250 ;Carrega R2 com 250.

DJNZ R2, $ ;Decrementa R2 até zerar.

DJNZ R1, VOLTA1 ;Decrementa R1 e salta até zerar.

DJNZ R0, VOLTA2 ;Decrementa R0 e salta até zerar.

RET ;Retorno de sub-rotina.

;SUBROTINA TEMPO=(250X250X240X2)uS= 30000000uS= 30s.

TEMPON: MOV R3, #240 ;Carrega R3 com 240.

VOLTA4: MOV R4, #250 ;Carrega R4 com 250.

VOLTA3: MOV R5, #250 ;Carrega R5 com 250.

DJNZ R5, $ ;Decrementa R5 até zerar.

DJNZ R4, VOLTA3 ;Decrementa R4 e salta até zerar.

DJNZ R3, VOLTA4 ;Decrementa R3 e salta até zerar.

RET ;Retorno de sub-rotina.

END ;Fim de programa

Exercício 6:

;========================================================================

; ================= ETEC Aristóteles Ferreira. ==================

; ======== Yhan Christian Souza Silva ===================

; ===== Programa\_01 exercicio6.asm ver. 0.0 data: 04/05/13. ========

;========================================================================

; Fazer um programa que contenha 3 chaves

;P2.7 = CHAVE GERAL

;P2.0 = LIGA E EXECUTA AÇÃO SEQUENCIA\_01

;P2.1 = LIGA E EXECUTA AÇÃO SEQUENCIA\_02

ORG 0000H ;Inicio de Programa

LOOP: MOV P0,#00H ;Limpo P0

MOV P2,#0FFH ;Preparo P2 para leitura

SEQUENCIA\_01: JB P2.0, SEQUENCIA\_02 ;Verifico o Acionamento do botão P2.0, pulo para rotina Sequencia\_02

JNB P2.7, GERAL ;Verifico se não foi acionado o botão P2.7, pulo para rotina GERAL

MOV A,#0FH ;Movo o dado 0Fh para o Acc

MOV P0,A ;Movo o dado de Acc para P0

LCALL TEMPO ;Sub-rotina de tempo

CPL A ;Complemento o valor de Acc

MOV P0, A ;Movo o dado de Acc para P0

LCALL TEMPO ;Sub-rotina de tempo

SEQUENCIA\_02:JB P2.1, GERAL ;Verifico o Acionamento do botão P2.1, pulo para rotina GERAL

JNB P2.7, GERAL ;Verifico se não foi acionado o botão P2.7, pulo para rotina GERAL

MOV A,#55H ;Movo o dado 55h para o Acc

MOV P0,A ;Movo o dado de Acc para P0

LCALL TEMPO ;Sub-rotina de tempo

CPL A ;Complemento o valor de Acc

MOV P0,A ;Movo o dado de Acc para P0

LCALL TEMPO ;Sub-rotina de tempo

GERAL: JB P2.7, SEQUENCIA\_01 ;Verifico o Acionamento do botão P2.1, pulo para rotina Sequencia\_01

CLR A ;Limpo Acc

MOV P0,A ;Movo o dado para P0

MOV P0,#00H ;Limpo P0

LCALL TEMPO ;Sub-rotina de tempo

SJMP LOOP ;Salto para LOOP

;SUBROTINA TEMPO=(250X250X2X2)uS= 250000uS=250ms

TEMPO: MOV R0, #2 ;CARREGA RO COM 2

VOLTA2: MOV R1, #250 ;CARREGA R1 COM 250

VOLTA1: MOV R2, #250 ;CARREGA R2 COM 250

DJNZ R2, $ ;DECREMENTA R2 ATÉ ZERAR

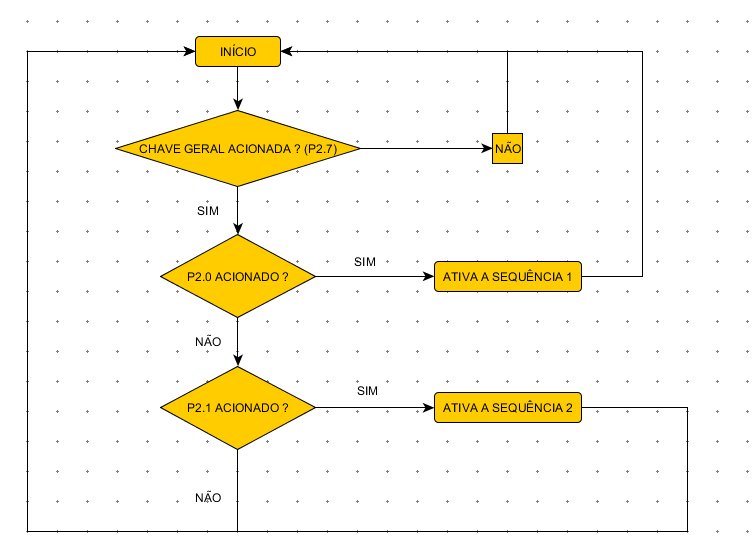
DJNZ R1, VOLTA1 ;DECREMENTA R1 E SALTA ATÉ ZERAR

DJNZ R0, VOLTA2 ;DECREMENTA RO E SALTA ATÉ ZERAR

RET ;RETORNO DE SUBROTINA

END ;FIM DO PROGRAMA

Fluxograma:



Exercício 7:

;========================================================================

; ================= ETEC Aristóteles Ferreira. ==================

; ======== Yhan Christian Souza Silva ===================

; ===== Programa\_01 exercicio7.asm ver. 0.0 data: 04/05/13. ========

; =======================================================================

;FAZER UM PROGRAMA QUE CONTENHA DUAS CHAVES:

;P2.0 = CHAVE GERAL

;P3.2 = EXECUTA SEQUENCIA\_01 E QUANDO PRESSIONADO EXECUTA SEQUENCIA\_02.

ORG 0000H ;Inicio de Programa

MOV P0,#00H ;Limpo P0

MOV P2,#0FFH ;Preparo P2 para leitura

LOOP: JB P3.2, GERAL ;Verifica o acionamento do botão P3.2, pulo para rotina GERAL

JNB P2.0, GERAL ;Verifico o Acionamento do botão P2.0, pulo para rotina GERAL

MOV A,#55H ;Movo para o Acc 55H

CPL A ;Complemento Acc

MOV P0,A ;Move o dado do Acc para o Port

JNB P3.2, $ ;Verifico se esta 0 o botão P2.0

CPL A ;Complemento Acc

MOV P0, A ;Movo o conteudo de Acc para P0

SJMP LOOP ;Salta para loop

GERAL: JB P2.0, LOOP ;Verifico o Acionamento do botão P2.1, pulo para rotina LOOP

CLR A ;Limpo Acc

MOV P0,A ;Movo o dado para P0

MOV P0,#00H ;Limpo P0

LCALL TEMPO ;Sub-rotina de tempo

SJMP LOOP ;Salto para LOOP

;SUB-ROTINA TEMPO

TEMPO: MOV R0, #2 ;CARREGA RO COM 2

VOLTA2: MOV R1, #250 ;CARREGA R1 COM 250

VOLTA1: MOV R2, #250 ;CARREGA R2 COM 250

DJNZ R2, $ ;DECREMENTA R2 ATÉ ZERAR

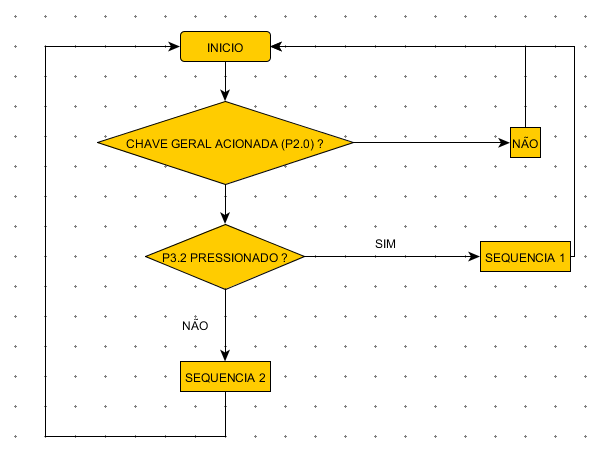
DJNZ R1, VOLTA1 ;DECREMENTA R1 E SALTA ATÉ ZERAR

DJNZ R0, VOLTA2 ;DECREMENTA RO E SALTA ATÉ ZERAR

RET ;RETORNO DE SUBROTINA

END ;FIM DE PROGRAMA

Fluxograma:



Exercício 8:

;========================================================================

; ================= ETEC Aristóteles Ferreira. ==================

; ======== Yhan Christian Souza Silva ===================

; ===== Programa\_01 exercicio8.asm ver. 0.0 data: 04/05/13. ========

;========================================================================

;FAZER UM PROGRAMA QUE CONTENHA TRÊS CHAVES:

;P2.3 = CHAVE GERAL

; P2.0 P2.7

; 0 0 Sequencia 0

; 0 1 Sequencia 1

; 1 0 Sequencia 2

; 1 1 Sequencia 3

ORG 0000H ;Inicio de Programa

MOV P0,#00H ;Limpo P0

MOV P2,#0FFH ;Preparo P2 para leitura

LOOP: ;Loop começo

SEQUENCIA\_0: JNB P2.0, SEQUENCIA\_2 ;Verifico se P2.0 está em nível lógico 0, pulo para Sequencia\_2

JNB P2.7, SEQUENCIA\_1 ;Verifico se P2.7 está em nível lógico 0, pulo para Sequencia\_1

JNB P2.3, GERAL ;Verifico se P2.3 está em nível lógico 0, pulo para GERAL

MOV A, #0FH ;Movo para Acc o dado 0fh

MOV P0, A ;Movo o dado de Acc para P0

LCALL SEQUENCIA\_3 ;Chamo sub-rotina Sequencia\_3

SJMP LOOP ;Salto para loop

SEQUENCIA\_1: JB P2.7, SEQUENCIA\_2 ;Verifico o acionamento do botão P2.7, pulo para Sequencia\_2

JNB P2.3, GERAL ;Verifico se P2.3 está em nível lógico 0, pulo para GERAL

MOV A,#0F0H ;Movo para Acc o dado f0h

MOV P0, A ;Movo o dado de Acc para P0

LCALL SEQUENCIA\_3 ;Chamo sub-rotina Sequencia\_3

SJMP LOOP ;Salto para loop

SEQUENCIA\_2: JB P2.0, SEQUENCIA\_3 ;Verifico o acionamento do botão P2.0, pulo para Sequencia\_3

JNB P2.3, GERAL ;Verifico se P2.3 está em nível lógico 0, pulo para GERAL

MOV A,#55H ;Movo o dado 55h para Acc

MOV P0, A ;Movo o dado de Acc para P0

LCALL SEQUENCIA\_3 ;Chamo sub-rotina Sequencia\_3

SJMP LOOP ;Salto para loop

SEQUENCIA\_3: JB P2.0, SEQUENCIA\_0 ;Verifico o acionamento do botão P2.0, pulo para Sequencia\_0

JB P2.7, SEQUENCIA\_0 ;Verifico o acionamento do botão P2.7, pulo para Sequencia\_3

JNB P2.3, GERAL ;Verifico se P2.3 está em nivel lógico 0, pulo para GERAL

MOV A,#0FFH ;Movo o dado FFh para Acc

MOV P0, A ;Movo o dado de Acc para P0

LJMP LOOP ;Salto para loop

GERAL: JB P2.3, LOOP ;Verifico o Acionamento do botão P2.3, pulo para rotina LOOP

CLR A ;Limpo Acc

MOV P0,A ;Movo o dado para P0

MOV P0,#00H ;Limpo P0

LJMP LOOP ;Salto para LOOP

END ;Fim de programa

Fluxograma: