

**UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE CIÊNCIAS E TECNOLOGIA
COORDENAÇÃO DE ENGENHARIA ELÉTRICA**

RELATÓRIO DE ESTÁGIO INTEGRADO

RUBENS GARCIA SILVA NERES

**Relatório apresentado à Coordenação de Estágios
de Engenharia Elétrica da UFPB como parte dos
requisitos necessários à obtenção do título de
Engenheiro Eletricista.**

MANAUS 12 DE DEZEMBRO DE 1998



Biblioteca Setorial do CDSA. Fevereiro de 2021.

Sumé - PB

1 - AGRADECIMENTOS

Meus agradecimentos vão para todos aqueles que de uma forma ou de outra, contribuíram nesta fase de minha vida, para o meu a engrandecimento profissional. Aqueles que não forem citados, não fiquem enciumados, pois sei a cota de contribuição de cada um de vocês. Agradeço a : Os engenheiros Francisco Carlos Ramos e Ricardo Toscano, aos Técnicos Nonato Libório, Elson, Sérgio e Francisco Costa, à secretária Rita de Cássia, à psicóloga Jozimary, ao Sr. Ítalo pelo exemplo de liderança, ao Sr. Plácido pelos conselhos profissionais e a todas as operadoras da linhas de montagem. Agradeço aos meus professores acadêmicos pelos ensinamentos teóricos em especial ao meu orientador José Gutemberg, aos amigos João Barbosa e Adail, aos meus pais e principalmente a Deus.

2 - APRESENTAÇÃO

O relatório descreverá um breve histórico da Panasonic da Amazônia S/A, apresentará uma introdução relatando superficialmente as atividades desenvolvidas durante o estágio, e em seguida descreverá detalhadamente todas as atividades realizadas neste estágio. Começaremos com uma descrição das atividades relacionadas à qualidade da empresa para depois enfocarmos as atividades relacionadas à engenharia de processo.

3 – HISTÓRICO DA PANASONIC

A Panasonic da Amazônia S/A situada à rua Matrinxã N° 1155, Distrito Industrial Marechal Castelo Branco na cidade de Manaus, Estado do Amazonas, foi criada em 01 de julho de 1981, da associação de duas empresa, a Springer e a National do Brasil, surgindo a Springer National da Amazônia S/A, tendo seu nome alterado para Panasonic da Amazônia em virtude da mudança da razão social em 1993.

Empresa do ramo eletro-eletrônico, tem hoje a montagem das seguintes linhas de produtos:

- 1- Televisor em cores
- 2- Forno de Microondas
- 3- Vídeo Cassete
- 4- Vídeo Câmera
- 5- Aparelhos de Áudio

A Panasonic, que empenha-se em fabricar produtos ambientalmente adequados com elevado nível de segurança e qualidade, implantou um Sistema de Qualidade de acordo com os requisitos da norma NBR ISO9002, tendo sido certificada em fevereiro de 1995. A Panasonic também procura manter a harmonia entre os avanços tecnológicos e a manutenção do meio ambiente, através do Sistema de Gestão Ambiental implantado na fábrica, tendo a mesma sido certificada de acordo com os requisitos da norma NBR ISO14000 em agosto de 1998. Empresa de origem japonesa procura manter nas suas atividades, a filosofia básica da administração de sua Matriz situada no Japão, a Matsushita Eletric, através dos slogans corporativos e da leitura dos sete princípios do grupo Matsushita.

4 – INTRODUÇÃO

Na primeira semana de contato com a empresa, fomos participar do programa de integração que todo novo funcionário é submetido. Várias palestras nos foi ministrada e os assuntos abordados foram:

ISO 9000

ISO 14000

SEGURANÇA DE TRABALHO

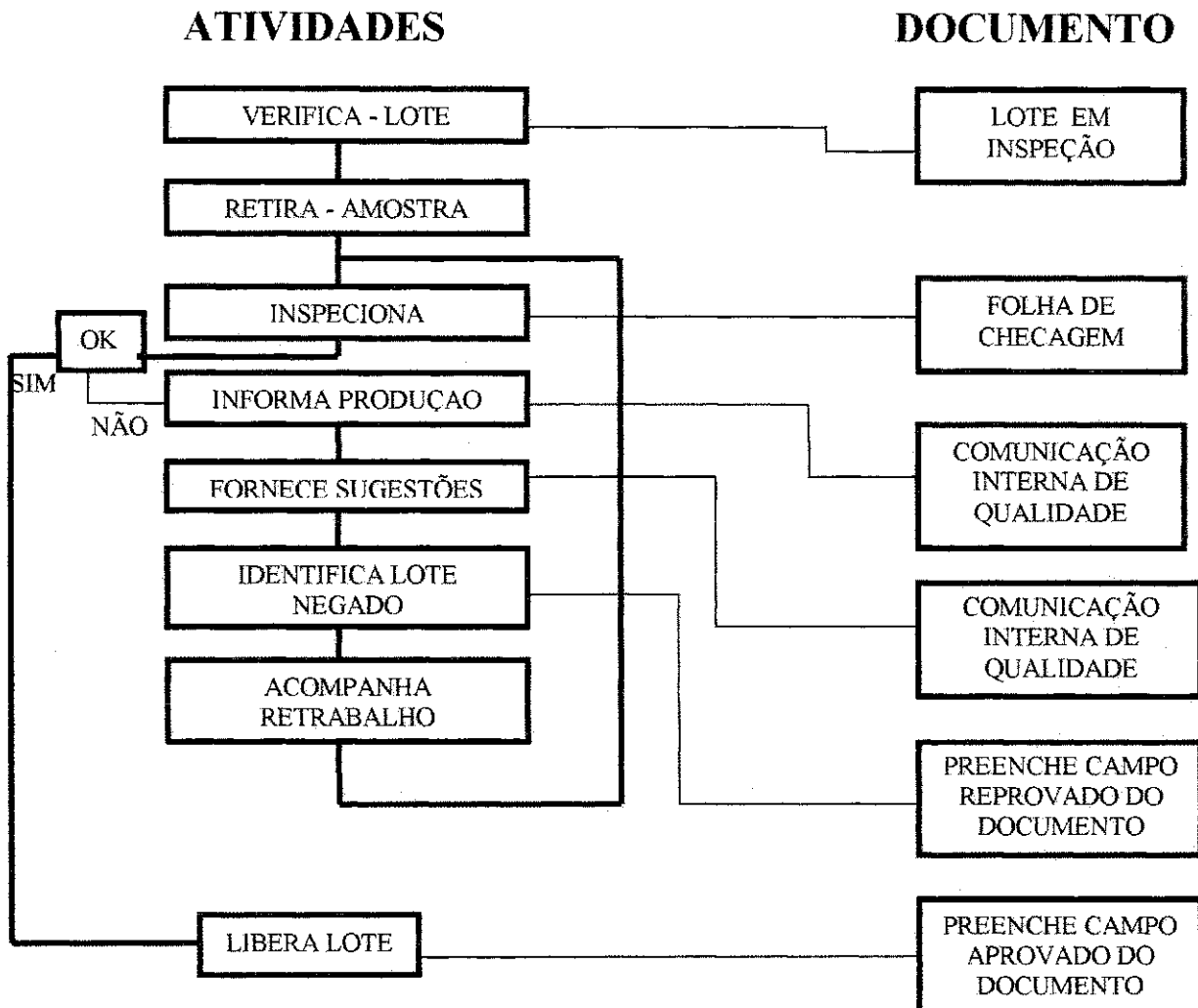
PLANO DE CARGOS E SALÁRIOS

Conhecemos cada setor e suas respectivas funções e responsabilidades. Seguidamente fomos conhecer cada mini-fábrica (forma como são chamado os setores da Panasonic) da empresa: Forno, Vídeo-Câmera, Áudio, TVC e IAC. Feito este reconhecimento e integração, fomos alocados no departamento de TV, onde iríamos desenvolver nossas atividades relativas ao estágio.

No departamento de TV fomos apresentados de maneira superficial a todo o processo produtivo. Posteriormente o descreveremos detalhadamente. Fomos informadas que nossas atividades seriam voltadas tanto para a área da Qualidade quanto para a engenharia, paralelamente, mais com maior direcionamento para a área de qualidade

5 – ATIVIDADES DE QUALIDADE

5.1 – Controle de Qualidade



A cada três lotes montados na TV um é inspecionado e identificado com o documento "Lote em Inspeção". O anterior e o posterior ficam em observação. Uma amostra aleatória é retirada do lote para inspeção. As análises são verificadas e os resultados são anotados na folha de checagem. Se as amostras estão conforme especificado, o lote é liberado e o campo aprovado do documento lote em inspeção é devidamente preenchido. Se houver não – conformidades, a produção é informada através de uma comunicação interna da qualidade, ações corretivas são tomadas e informadas com o mesmo documento, o campo reprovado do documento lote em inspeção é preenchido e o retrabalho é realizado e acompanhado.

5.2 – Aferição / Calibração de Equipamentos

Uma atividade de grande importância dentro da empresa é a aferição e calibração dos equipamentos de medição inspeção e ensaio. De posse de uma lista mestre com todos os equipamentos eletro-eletrônicos do departamento, sua localização e situação de aferição / calibração, ficamos sabendo quais equipamentos estão necessitando de atualização. A calibração é feita da seguinte forma: Em um manual de procedimentos de aferição / calibração de equipamentos, encontramos todas as informações necessárias para realizar este trabalho. Tomaremos como exemplo um osciloscópio da marca TRIO localizado na linha de inspeção final 03 (IF03). No manual diz que necessitamos de um osciloscópio padrão com certificado de calibração, um gerador de sinais, pontas de prova e ponteiras. O procedimento é fornecer através do gerador de sinais vários níveis de tensão medidos em nosso processo produtivo, ligar o osciloscópio padrão, o osciloscópio que se quer aferir e o gerador de sinais em paralelo. Ajusta-se os níveis de tensão no osciloscópio padrão, que teoricamente está perfeitamente calibrado e faz-se a leitura com o aparelho que se está aferindo. Realizamos no mínimo três replicações a intervalos de dez minutos, e através de cálculos de adequação e exatidão, damos um laudo favorável, ou não favorável ao uso do equipamento no processo, em um relatório de calibração. Estes cálculos são mostrados numa página em anexo com um modelo de um relatório, onde a incerteza da medição foi calculada com um nível de confiança de 95%. Para este caso específico o osciloscópio em teste foi considerado fora das especificações de adequação e exatidão. O próximo passo é a aferição/calibração do equipamento externamente. Como a não conformidade continuou, abrimos o equipamento, fazemos um estudo detalhado do diagrama elétrico do mesmo e tentamos calibrá-lo internamente, com ajuste de potenciômetros, bobinas, etc.. Se a tentativa de colocá-lo dentro dos padrões for bem sucedida, nós o identificamos com etiquetas, e o colocamos de volta ao processo. De acordo com os resultados a identificação se dá com as etiquetas mostradas na página seguinte. Neste exemplo o equipamento foi considerado não conforme. O mesmo recebe etiqueta de identificação apropriada, seu relatório é arquivado em pastas (seguindo as normas ISO9000) e o equipamento é acondicionado em local apropriado. Da mesma forma, se o laudo fosse favorável ao uso no processo, tomaríamos os mesmos procedimentos.

Panasonic Depto TV
N° Controle equipamento:
Calibrado em: / /
Próxima calibração: / /
Responsável:

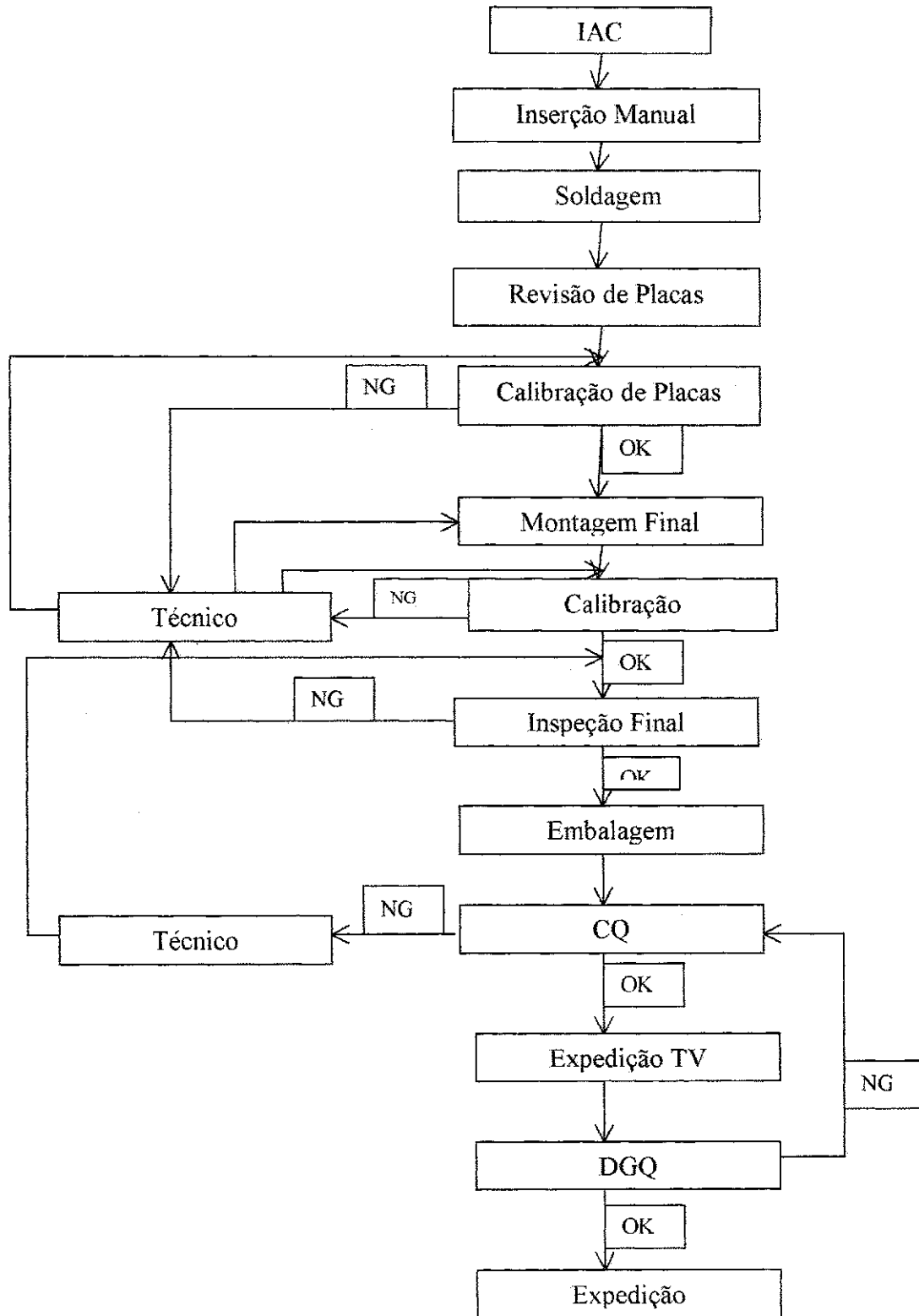
Panasonic Depto TV
N° Controle equipamento:
Não requer aferição / calibração
Periódica
Responsável :

Panasonic Depto TV
N° Controle equipamento: 00
Equipamento fora de uso
Responsável:

Panasonic Depto TV
Nr. Cont. Equipeto:
INSTRUMENTO DE USO RESTRITO
Resp.:

6 – ATIVIDADES DE ENGENHARIA DE PROCESSO

A fábrica de TV tem seu processo produtivo realizado seguindo o máximo que possível o fluxograma abaixo:



6.1 – Processo produtivo

6.1.1 – IAC

Todos os modelos de TV's produzidos na Zona Franca de Manaus pela Panasonic da Amazônia S/A são projetados conjuntamente em São José dos Campos, Malásia, EUA e Japão. A placa chega ao IAC (departamento de Inserção Automática) limpa, sem qualquer componente. Apenas furos, serigrafia com a posição de cada componente e uma lista de peças destes componentes com seus respectivos códigos. A partir da lista, a placa irá passar por processos dentro do IAC até sua saída para o departamento de TV. Estes processos são:

- 1- Programação
- 2- Inserção de ilhós
- 3- Inserção de fios jumpers
- 4- Inserção de componentes axialmente (Horizontal)
- 5- Inserção de componentes radialmente (Vertical)
- 6- Revisão
- 7- Inserção de SMD
- 8- Revisão

6.1.1.1 - Programação

Com a lista de peças e a seqüência de inserção (pré determina), a placa é introduzida numa máquina senquenciadora de coordenadas que contém um programa que grava a seqüência, o componente e suas dimensões, à medida que um câmara faz a leitura sobre a placa. Cada modelo possui um programa diferente, pois os componentes podem mudar de um modelo para outro. Nos processos 2, 3, 4, 5 e 7 temos um programa para cada máquina de inserção, sendo que nos processos 4 e 5 o processo é subdivido em sub-processos para uma maior rapidez de produção.

6.1.1.2 - Inserção de Ilhós

Neste processo realizado na máquina Panasert E são inseridos ilhós de 1,6 e 2,5 milímetros de diâmetro (somente TV) para aumentar a fixação de componentes no chassis (flyback, capacitores eletrolíticos, etc.).

Sensores detectam a presença do componente e feixes de luz emitidos de baixo para cima da placa acusam se o local está apto, ou não apto a receber o ilhós. Se o feixe passa pelo furo, o local está apto, se não, é porque algo está obstruindo o local. Ocorrendo isto, ou uma inserção indevida, a máquina para de funcionar e aciona um alarme luminoso. Então um operador corrige o problema e dá continuidade ao processo. Se faltar componente para alimentar a máquina, sensores detectam esse fato e o operador a alimenta .

6.1.1.3 - Inserção de jumpers

A máquina Panasert JVK insere fios jumpers de comprimentos diferentes e mesmo diâmetro nos eixos x e y, e na seqüência determinada pelo programa. O Programa determina o comprimento do fio, ordena o corte e na parte inferior da placa um mecanismo faz a dobra dos terminais frontalmente (um de frente para o outro) e com uma certa inclinação. Se ocorre uma inserção incorreta, ou falta fio no carro alimentador, a máquina para e o operador entra em ação para dá continuidade ao processo. Os fios são inseridos todos de uma vez no eixo x para depois serem inseridos no eixo y.

6.1.1.4 - Inserção de componentes radialmente

Três Panasert AVK realizam esse processo. A divisão de trabalho ocorre para dá mais número à produção, mas ambas realizam as mesmas tarefas. Aqui, resistores, diodos e bobinas são inseridos na placa . O Programa determina a ordem dos componentes a serem inseridos, o corte e a dobra dos terminais nas dimensões especificadas. Para este caso os terminais são dobrados de frente um para o outro e com uma inclinação de 45° com relação ao chassis, para que os mesmos não caiam ao serem transportados. À entrada da mesa de inserção, sensores indicam a existência ou não de componentes, enquanto outros indicam se o local onde determinado componente será inserido está apto, ou não de recebe-lo. Isso é feito com a emissão de um feixe de luz, que é emitido sob a placa. Se o feixe passa pelo furo e chega ao sensor, o local está apto a receber, se não, a máquina para e não insere o componente. Outro mecanismo da máquina detecta se o componente está, ou não partido, simulando um curto

em seus terminais. Se o curto é observado o componente está OK, se não, o mesmo encontra-se partido e então um alarme luminoso é acionado. À entrada da máquina outros sensores detectam quando a fita com os componentes deve ser trocada.

6.1.1.5 - Inserção de componentes radialmente

Duas Panasert RH realizam este processo e o princípio de funcionamento é o mesmo das máquinas de inserção axial, com a diferença de que neste caso os componentes são inseridos verticalmente sobre a placa. Esses são capacitores, transistores, pinos, etc.

6.1.1.6 – Revisão

O processo seguinte é uma revisão feita por uma pessoa. Com um chassis modelo, com todos os componentes de inserção automática inseridos, este revisor compara com as placas que estão saindo do processo e verifica a sua conformidade. Se ocorre a falta de algum componente, ou se algum foi inserido incorretamente, verifica-se se mais ocorrências irão acontecer, para saber se o erro foi aleatório, ou se o problema está em alguma máquina.

6.1.1.7 - Montagem SMD

Última etapa pela qual o chassis de TV passa dentro do departamento de inserção automática. SMD (surface mounting devices), montagem de dispositivos superficialmente, é a tecnologia de inserção automática de componentes de dimensões reduzidas e que são colados na superfície cobreada da placa e não inseridos em furos. Numa primeira máquina, uma gota de um adesivo é colocado exatamente onde o componente será colado e uma segunda, filma o componente através de uma mini-câmera à entrada da mesa de inserção, checa suas dimensões e o tipo de componente. Se tudo estiver conforme o programa ordena, o componente é colado sobre a gota de adesivo. A terceira máquina do processo complementa esse trabalho. Passada essa fase, o chassis passa por uma máquina aquecedora (forno) para fins de cura do adesivo. A temperatura de aquecimento é controlada e não superior a 140° C para não provocar danos aos

componentes (alterar suas características elétricas). Em seguida uma revisora confere possíveis falhas. Terminada todas essas fases as placas são armazenados em magazines e transportados ao departamento de TV para início do processo produtivo de inserção manual.

6.1.2 - Linha de Placas de Circuito Impresso (CI's)

Depois que a placa passa por todo processo de inserção automática, já na fábrica de TV, irá ser submetida ao processo de inserção manual de componentes. A partir de uma lista de peças dos componentes, faz-se uma checagem se todos os componentes de inserção automática foram inseridos, se os de inserção manual concordam em código, número de vez que é utilizado, e posição com a serigrafia impressa na placa. Conferida esta parte, soma-se o número de componentes que serão inseridos manualmente (incluindo os repetidos) e divide-se pelo número de operadoras, sendo que as peças maiores (transformadores, dissipadores, flayback, etc.) são deixadas por último. A divisão é feita para se ter uma idéia de quantos componentes por operadora teremos em cada posto. Faz-se um estudo detalhado sobre quais componentes devem ser inseridos primeiro, para que inserções futuras não sejam prejudicadas por inserções anteriores. Por exemplo: quando for inserido uma nova peça, uma outra não seja empurrada entortando-se seus terminais. Cada operadora também contribui com sua opinião, informando quais peças do seu posto, proporcionariam melhor desempenho, se fossem inseridas primeiro, ou depois. Esta tarefa desenvolvida pela operadora, é feita com as duas mãos alternadamente, e a ordem com que a mesma inicia seu trabalho varia de operadora para operadora. O importante é que uma nova inserção não prejudique inserções anteriores, como citado anteriormente, forçando componentes já inseridos. Após esse estudo preliminar, instruções de trabalho (seqüência a ser seguida pela operadora) são elaboradas para serem seguidas.

Feita a inserção dos componentes temos um posto onde é feita uma revisão e onde um chassis (como são chamadas as placas depois de inseridos todos os seus componentes) padrão serve como padrão para a operadora. Seguindo o processo, os chassis passam por uma máquina de soldagem (estanhadeira) para soldagem dos terminais dos componentes que ainda estão soltos. Como a máquina não é perfeita, ocorre curtos por solda ou mesmo curto por solda fria. Então três postos existem para uma

revisão destas soldas e retoque de algumas, como no caso dos terminais do flyback. Nesta fase são colocadas algumas peças como fios jumpers, cabo de força, fio de alta tensão do flyback, etc.

O passo seguinte é a calibração destes chassis em termos de PRESET (seta todos os dados de fábrica no IC de memória do TV), CAG, AFT, SUB-BRILHO, SUB-MATIZ, SUB-COR E SUB-CONTRASTE. Se algum defeito tipo: falta de componente, ou curto por solda é detectado pelas operadoras, o chassis volta para o posto de revisão para correção. Se o defeito não é descoberto logo nas operadoras, estes seguem para o posto dos técnicos em eletrônica para serem concertados. Caso haja necessidade de nova calibração o chassis retorna ao ponto onde precisa ser calibrado, se não, este é armazenado em carrinhos transportadores para entrarem na fase de acabamento.

6.1.3 - Linha de Acabamento (Inspeção Final)

Depois que os chassis passam pelas calibrações, estes são conduzidos ao acabamento. Nesta fase do processo, os chassis são inseridos nos gabinetes, são inseridos: bobinas desmagnetizadoras, alto-falantes e cinescópios. Os aparelhos entram numa fase de aquecimento para poderem sofrerem as últimas calibrações, que são ajuste de pureza, ajuste de white balance e ajuste da tensão de cutoff. A tampa traseira é colocada no aparelho e os seguintes testes são realizados: teste de isolamento, ruptura, e teste do funcionamento de teclas. O passo seguinte é a embalagem do aparelho, colocação dos acessórios (manual de instrução do usuário, controle remoto, pilhas e um balum) e lacre da embalagem. Seguidamente temos o empilhamento em paletes, inspeção de qualidade e armazenamento para expedição.

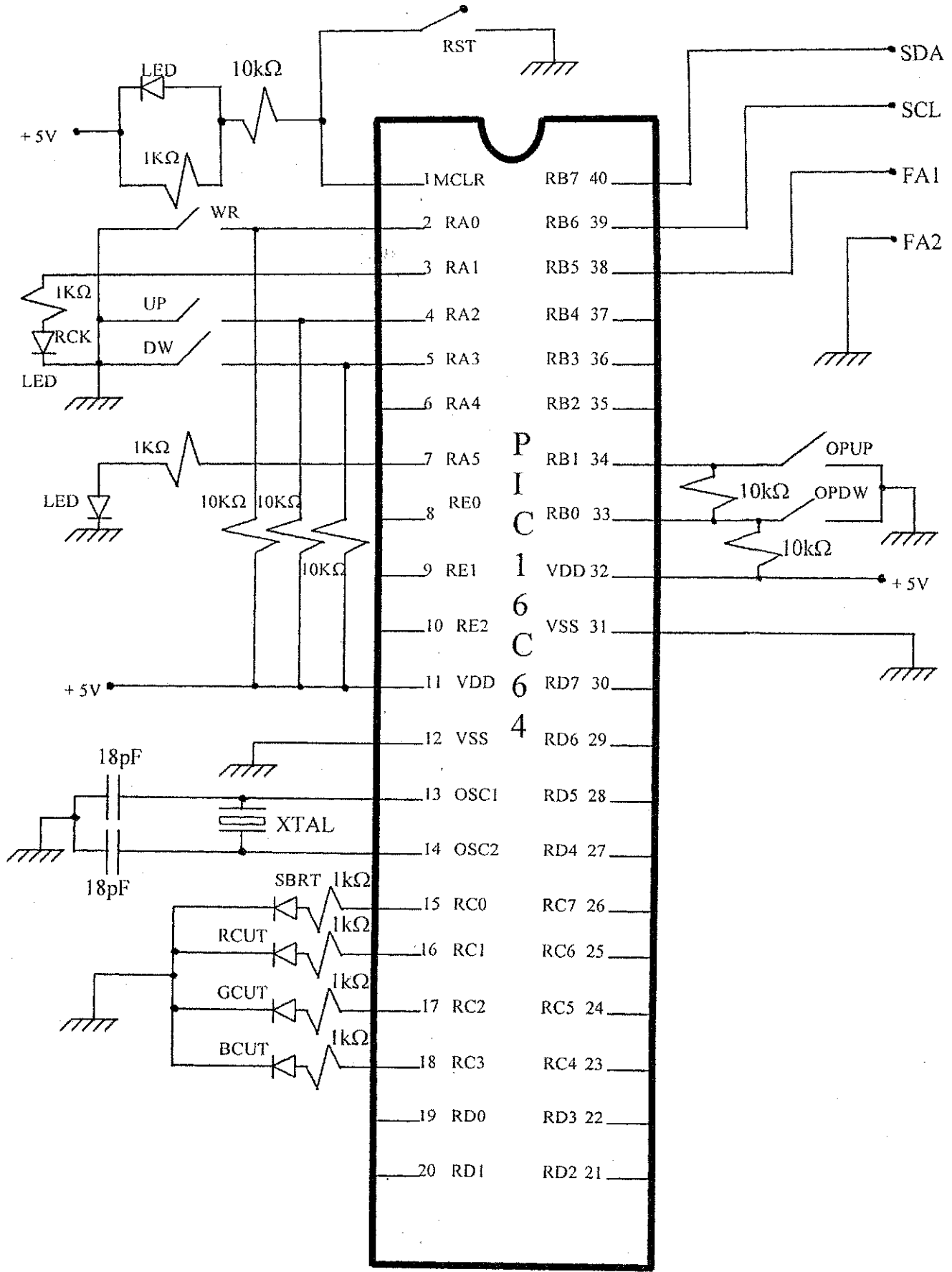
6.1.4 – Outras Atividades

Ainda referente a engenharia de processo foi realizada a instalação de três bases de calibração do chassis MX – 5 dos modelos TC - 20A8 e TC20B8. Este trabalho consistiu na instalação dos jig's com as agulhas localizadas nas posições onde deve haver contato com a placa, para que os ajustes possam ser feitos. Numa determinada base fizemos conexões de ponteiras de osciloscópios, de frequencímetros, de multímetros, de osciladores, etc.. Através de um controlador de programas, selecionamos

automaticamente, o teste ou o ajuste que queremos realizar. Estes testes ou ajustes podem ser de : PRESET, VCO, CAG, VÍDEO, SUB – COR, SUB – BRILHO, etc.. O controlador de programas é um equipamento capaz de controlar quatro modelos diferentes, contendo cada modelo 25 passos. Em cada passo pode-se programar 25 pontos de saída , 8 pontos de entrada, um timer variando de 0.1 a 9.99 segundos, atenuações e saída de stopper. Este controlador possui uma memória EEPROM e uma CPU Z80 com um clock de 4 Mhz .

6.1.5 – Mini-Projeto de Engenharia

Agora iremos detalhar nosso projeto de engenharia que nos foi solicitado. O mesmo consiste na automatização do ajuste da tensão de corte do cinescópio, ou como é mais comumente conhecida tensão de cutoff. Hoje o ajuste é feito através do controle remoto, entrando-se no modo fábrica do aparelho e ajustando-se os valores dos registros do IC de memória, e do IC de Vídeo Chroma Jungle (VCJ), de acordo com os valores especificados pelo fabricante, no manual de calibração. Com o projeto iremos setar estes valores diretamente nestes IC's mencionados anteriormente, reduzindo significativamente o tempo gasto neste processo. Isso será feito através do uso do microcontrolador PIC16C64 da Microchip. Através de programação em assembly, programamos o microcontrolador para tomar de conta do barramento IIC. O protocolo IIC de comunicação é atualmente utilizada em Vídeos, TV's e outros equipamentos eletrônicos devido a sua simplicidade. Basicamente necessita-se de um barramento de dados (SDA) e um de clock (SCL), para realizar-se toda comunicação. Toda vez que há um pulso de clock, é enviado um bit. A informação é recebida a partir de um bit de partida, o qual inicializa a emissão de dados. Este protocolo segue a seguinte orientação: envia-se o endereço do componente que se quer comunicar, envia-se o endereço de alocação na qual se deseja gravar ou ler dados, e por fim envia-se o dado . Após todos estes dados serem enviados , manda-se um stop bit para indicar que a comunicação já foi feita e que a informação foi enviada. No nosso caso iremos nos comunicar com o VCJ e com a Memória. A figura mostrada na página seguinte mostra a estrutura do nosso controlador com a visão dos seus 40 pinos e o respectivo diagrama elétrico do projeto. O programa que comanda todas as ações do PIC segue em anexo e as chaves funcionam com descrito abaixo:



OPUP – Seleção de opção de ajuste para cima
OPDW – Seleção de opção de ajuste para baixo
UP – Incrementa o valor do registro
DW – Decrementa o valor do registro
WR – Grava no registro o valor ajustado
RST – Inicializa o programa
Led sbrt – Indica que o ajuste é no registro de sub-brilho
Led r-cut – Indica que o ajuste é no registro de r-cutoff
Led g-cut – Indica que o ajuste é no registro de g-cutoff
Led b-cut – Indica que o ajuste é no registro de b-cutoff
Led rck – Indica que houve falha na comunicação

Com o manual de calibração tiramos a informação de qual valor de tensão de cutoff deve-se ajustar e em qual ponto da placa devemos tomar com referencia para efetuar os ajustes:

Para 14" temos 140 Vac
Para 20" temos 160 Vac
Para 29" temos 160 Vac

Através do osciloscópio conectado no ponto indicado, que para o modelo TC-20A8 é no coletor do transistor Q032 da placa Y, ajustamos a tensão de cutoff através das chaves UP e DW no valor especificado.

7 – CONCLUSÃO

É inegável que durante esses seis meses em que o nosso relatado estágio se desenvolveu, uma gama muito grande de conhecimentos nos foi proporcionada, contribuindo infinitamente para aumentar nossos conhecimentos teóricos, repassados pelos nossos mestres acadêmicos, e para desenvolver conhecimentos práticos, dando-nos uma experiência profissional inicial, para a partir deste suporte conseguirmos almejar nossos objetivos de vida. Além da experiência profissional, também adquirimos experiência de vida, aprendendo a lidar com todo tipo de pessoas, como nos comportar diante de situações de pressão e de dificuldades. Infelizmente em qualquer lugar nem tudo são maravilhas: sempre vai haver pontos negativos, frustrantes e decepcionantes, mas nada que um bom humor e um bom profissionalismo não possam superar. Enfim com este aprendizado adquirido na Panasonic poderemos nos tornar profissionais competentes e aplicar esses conhecimentos em benefício de nós mesmos e da empresa.

ANEXOS

Panasonic

Departamento TV

Relatório de Calibração

Nº 036/98

NOME DO EQUIPAMENTO: OSCILOSCÓPIO	FAIXA NOMINAL: 0 - 40 Vpp
FABRICANTE: KENWOOD MODELO: VCS-5135	MENOR DIVISÃO: 0.4mV
Nº CONTROLE: BTV00148 Nº SÉRIE: 10200762	LOCAL DE USO: RESERVA
CONDIÇÕES DO TESTE: TEMP. 27 °C UMIDADE 74 %	TOLERÂNCIA DO PROCESSO(A): ±
DATA : 09/09/1998	PADRÃO UTILIZADO NA CALIBRAÇÃO: 1- OSCILOSCÓPIO PANASONIC VP-5020A; 2- AUDIO SIGNAL GENERATOR LEADER LAG-120B
CALIBRADO POR: RUBENS GARCIA SILVA NERES	CERTIFICADO Nº: 1- BALITEK SP0549/98 2- FUCAPI 879/96

CH1

Vr(Vpp)	Range	R1	R2	R3	Média	d	DP	ϵ	lm	lh	lt1
0.60	20mV	0.76	0.76	0.76	0.76	0.0200	0.00	0.0240	0.0000	0.0003	0.0208
1.00	20mV	0.96	0.96	0.96	0.96	0.0400	0.00	0.0300	0.0000	0.0010	0.0410
1.20	20mV	1.15	1.15	1.15	1.15	0.0500	0.00	0.0360	0.0000	0.0012	0.0512
2.20	50mV	2.15	2.15	2.15	2.15	0.0500	0.00	0.0660	0.0000	0.0022	0.0522
2.50	50mV	2.45	2.45	2.45	2.45	0.0500	0.00	0.0750	0.0000	0.0025	0.0525
2.60	50mV	2.55	2.55	2.55	2.55	0.0500	0.00	0.0780	0.0000	0.0026	0.0526
2.90	0.1V	2.80	2.80	2.80	2.80	0.1000	0.00	0.0870	0.0000	0.0029	0.1029
3.00	0.1V	2.90	2.90	2.90	2.90	0.1000	0.00	0.0900	0.0000	0.0030	0.1030
3.30	0.1V	3.20	3.20	3.20	3.20	0.1000	0.00	0.0990	0.0000	0.0033	0.1033

CH2

Vr(Vpp)	Range	R1	R2	R3	Média	d	DP	ϵ	lm	lh	lt 2
0.80	20mV	0.76	0.76	0.76	0.76	0.0400	0.00	0.0240	0.0000	0.0008	0.0408
1.00	20mV	0.92	0.92	0.92	0.92	0.0800	0.00	0.0300	0.0000	0.0010	0.0810
1.20	20mV	1.10	1.10	1.10	1.10	0.1000	0.00	0.0360	0.0000	0.0012	0.1012
2.20	50mV	2.00	2.00	2.00	2.00	0.2000	0.00	0.0660	0.0000	0.0022	0.2022
2.50	50mV	2.30	2.30	2.30	2.30	0.2000	0.00	0.0750	0.0000	0.0025	0.2025
2.60	50mV	2.50	2.50	2.50	2.50	0.1000	0.00	0.0780	0.0000	0.0026	0.1026
2.90	0.1V	2.70	2.70	2.70	2.70	0.2000	0.00	0.0870	0.0000	0.0029	0.2029
3.00	0.1V	2.80	2.80	2.80	2.80	0.2000	0.00	0.0900	0.0000	0.0030	0.2030
3.30	0.1V	3.10	3.10	3.10	3.10	0.2000	0.00	0.0990	0.0000	0.0033	0.2033

CÁLCULOS

$\Delta \geq lt$ e $Vr - \epsilon + lh \leq Média \leq Vr + \epsilon - lh$					
Vr(V)	Δ (V)	Adeq.Ch1	Exat.Ch1	Adeq.Ch2	Exat.Ch2
0.80	0.08	ok	ok	ok	ng
1.00	0.05	ok	ng	ng	ng
1.50	0.2	ok	ng	ok	ng
2.20	0.1	ok	ok	ng	ng
2.50	0.1	ok	ok	ng	ng
2.60	0.1	ok	ok	ng	ng
2.90	0.1	ng	ng	ng	ng
3.00	0.1	ng	ng	ng	ng
3.30	0.1	ng	ng	ng	ng

CALIBRADOR OK

LEGENDA:

Vr - Valor de referência lh - Incerteza herdada do padrão
R - Replicações lt - Incerteza total
DP - Desvio padrão ϵ - Precisão do instrumento medido
d - Desvio relativo ou erro Δ - Tolerância exigida no processo.
lm - Incerteza medida lt - incerteza total ($d+lm+lh$)

PROCEDIMENTOS:

OS PROCEDIMENTOS ESTÃO BASEADOS NO MANUAL DE CALIBRAÇÃO E NO MANUAL DE INSTRUÇÃO DO EQUIPAMENTO.

OBS:

DE ACORDO COM OS RESULTADOS O EQUIPAMENTO ENCONTRA-SE FORA DE CONDIÇÕES DE USO.

RESULTADO DA CALIBRAÇÃO:

APROVADO PARA USO GERAL
 APROVADO COM RESTRIÇÃO
 REPROVADO

RESPONSÁVEL

VISTO

```

list p=16c64a, c=132
;Este programa implementa um controlador i2c, equipamento
;destinado a efetuar a calibracao de sinais em televisores.
;Consiste em variar o controle de sinal diretamente no ic de
;bcj, atraves do barramento i2c ate conseguir o ajuste e
;gravar este dado de ajuste na memoria 24c01.

```

```

; Panasonic da Amazonia S/A
; Projetista: Francisco carlos Ramos de Souza &
; Rubens Garcia Silva Neres

```

```

; cut-off
; MX-5

```

```

;*****
; definicao de registros
;*****

```

```

;
status equ 3h ;registro de status
port_a equ 5h ;porta(5), usada para chaves up, down, wr
port_b equ 6h ;portb(6), usada p/dados e clock
port_c equ 7h ;portc(7), usada p/acender leds
eeprom equ 20h ;buffer de bits
bycnt equ 21h ;contador de bytes p/modo leitura
addr equ 22h ;contador de enderecos
datai equ 23h ;registrador de dados de entrada
datao equ 24h ;registrador de dados de saida
slave equ 25h ;endereço do slave
txbuf equ 26h ;buffer de transmissao
count equ 27h ;contador de bits
bcount equ 28h ;contador de bytes
loops equ 29h ;contador do loop de atraso
loops2 equ 2ah ;contador do loop de atraso
addrm equ 2bh ;sub end p/leitura/escrita na memoria
slavem equ 2ch ;end. slave da memoria
cpcount equ 2eh ;contador de opcoes de ajuste
tempstr equ 2fh ;contadorde atraso
co_sw equ 30h ;chave de cut-Off
tempdt equ 31h ;registro temporario de dados

```

```

;*****
; definicao de bits
;*****

```

```

;
di equ 7 ;bit de entrada da eeprom
do equ 6 ;bit de saida da eeprom
sdata equ 7 ;linha de dados (portb 13)
sclk equ 6 ;linha de clock (portb 12)
ackf equ 1 ;led de rec. de falha (porta 18)
z equ 2

```

```

;*****
; Power up
;

```

```

org 01ffh ;seta vetor de reset
org 000h ;
movlw b'00001101' ;seta port_a como saida
tris port_a ;
bcf port_a,1 ;seta bit1 em L
bcf port_a,5 ;
movlw b'11011111' ;seta port_b
tris port_b ;
bsf port_b,5 ;
movlw 0x00 ;seta port_c como saida
tris port_c ;
clrf port_c ;apaga todos os leds
clrf opcount ;zera contador de opcoes

```

```

;*****

```

```

; opcao de ajuste
;*****
opcao
cpup    clr    port_c      ;apaga todos os leds
        btfs   port_b,1    ;checa sw opup
        goto   opinc       ;se acionada, vai para opinc
opdown  btfs   port_b,0    ;checa sw opdown
        goto   opdec       ;se acionada, via para opdec
        goto   opadj       ;
;-----
opinc   call   WAIT1       ;espera
        btfs   port_b,1    ;debouce da chave opup
        goto   opinc       ;
        movlw  0x03        ;maximo 03, 4 opcoes de ajuste
        subwf  opcount,w   ;
        btfs   status,z    ;
        incf   opcount,1   ;incrementa contador de opcoes
        goto   opadj       ;
opdec   call   WAIT1       ;espera
        btfs   port_b,0    ;debouce da chave opdown
        goto   opdec       ;
        movlw  0x00        ;minimo zero, 4 opcoes de ajuste
        subwf  opcount,w   ;
        btfs   status,z    ;
        decf   opcount,1   ;
        goto   opadj       ;
;-----
opadj   movlw  0x00        ;testa contador para sbprt
        subwf  opcount,w   ;
        btfs   status,z    ;
        goto   sbprt_sw    ;vai para ajuste de sbprt
        movlw  0x01        ;testa contador para rcoff
        subwf  opcount,w   ;
        btfs   status,z    ;
        goto   rcoff_sw    ;
        movlw  0x02        ;testa contador para gcoff
        subwf  opcount,w   ;
        btfs   status,z    ;
        goto   gcoff_sw    ;
        movlw  0x03        ;testa contador para bcoff
        subwf  opcount,w   ;
        btfs   status,z    ;
        goto   bcoff_sw    ;
        goto   opcao       ;
;*****
sbprt_sw
        movlw  b'11011111' ;seta porta b
        tris   port_b      ;
        bsf    port_c,0    ;acende led ind. SBRT
;-----
        movlw  0xa0        ;seta slave como a0h
        movwf  slave       ;memoria
        movlw  0x1c        ;seta sub end como 1ch
        movwf  addr        ;r-drive
        movlw  0x80        ;dado inicial de r-drive
        movwf  datao       ;80h
        call   FA1         ;coloca FA1 em L
        call   WRBYTE      ;chama rotina de escrita
        call   WAIT1       ;espera
        movlw  0x1d        ;seta sub end com 1dh
        movwf  addr        ;b-drive
        call   WRBYTE      ;chama rotina de escrita
        call   WAIT1       ;
        movlw  0x19        ;seta sub-end como 19h
        movwf  addr        ;
        movlw  0x00        ;dado inicial de r-cutoff
        movwf  datao       ;e b-cutoff

```

```

call    WRBYTE      ;
call    WAIT1       ;
movlw   0x1b        ;set sub end como sendo 1bh
movwf   addr        ;
call    WRBYTE      ;
call    WAIT1       ;
movlw   0x1a        ;seta sub end como sendo 1ah
movwf   addr        ;
movlw   0x25        ;dado inicial de g-cutoff 25h
movwf   datao       ;
call    WRBYTE      ;
call    WAIT1       ;
movlw   0x1f        ;cut-off sw
movwf   addr        ;
movlw   0x40        ;dado inicial de cutoff_sw 40h
movwf   co_sw       ;registro de cut-off sw
movwf   datao       ;coloca dado de cutoff em datao
call    WRBYTE      ;chama rotina de escrita
call    WAIT1       ;espera
;bsf   port_b,5    ;//
;call  WAIT        ;//
;call  WAIT        ;//
;call  WAIT        ;//
;call  WAIT        ;//
;call  WAIT        ;//
;call  WAIT        ;//
;call  WAIT        ;//
;call  WAIT        ;//
;call  WAIT        ;//
;call  WAIT        ;//
movlw   0xba        ;seta slave como bah
movwf   slave       ;vcj
movlw   0x10        ;vou falar c/ end 10h no vcj
movwf   addr        ;
movf    co_sw,w     ;\\
movwf   datao       ;carrega co_sw em datao
call    FA1         ;\\
call    WRBYTE      ;
call    WAIT1       ;escreve datao no vcj
movlw   0x13        ;seta sub end como 13h
movwf   addr        ;service sw
movlw   0x08        ;fecha o vertical
movwf   datao       ;
call    WRBYTE      ;escreve 08h no reg do vcj
call    WAIT1       ;espera

```

```

movlw   0xa0        ;seta slave como sendo a0h
movwf   slavem      ;memoria
movlw   0x16        ;seta end da memo como 16h
movwf   addrm       ;SUB-BRILHO memoria
movlw   0xba        ;seta slave como sendo bah
movwf   slave       ;vcj (escrita)
movlw   0x0a        ;seta sub end. como 0ah
movwf   addr        ;SUB-BRILHO vcj
call    FA1         ;
call    READ        ;chama sub rotina de leitura
call    WAIT1       ;
;bsf   port_b,5    ;
movf    data1,w     ;copia data1 em
movwf   datao       ;datao
goto   ajuste      ;

```

ajuste

```

movlw   b'11011111' ;seta port_b
tris   port_b       ;
btfss  port_b,0     ;testa bit 0 da porta b
goto   opcao        ;se 0, vai p/opcao
btfss  port_b,1     ;testa bit 1 da porta b

```



```

goto      opcao      ;se 0 vai p/opcao
btfss    port_a,0    ;le sw_memo skip se nao zero
goto      sw_memo    ;se zero vai para sw memo
btfss    port_a,2    ;le sw_inc skip se nao zero
goto      sw_inc     ;se zero vai para sw_inc
btfss    port_a,3    ;le sw_dec skip se nao zero
goto      sw_dec     ;se zero vai sw_dec
sw_inc   goto      ajuste ;
btfsc    port_b,5    ;
call     FA1         ;
call     WAIT3      ;
movlw    0xff        ;maximo 255d para sub-brilho
subwf    datao,w     ;
btfsc    status,z   ;
goto     max        ;
incf     datao,1    ;incrementa dado
max      nop        ;desab. o uP e habil. o control.
movlw    0x20       ;
subwf    datao,0    ;
btfss    status,0   ;
goto     zerobm     ;
movlw    0x20       ;
subwf    datao,1    ;
zerobm   call     WRBYTE ;escreve no ic vcj
movlw    0x20       ;
addwf    datao,1    ;
sw_dec   goto     ajuste ;
btfsc    port_b,5   ;
call     FA1        ;
call     WAIT3      ;
movlw    0x00       ;
addwf    datao,w    ;
btfsc    status,z   ;
goto     min        ;
decf     datao,1    ;
btfsc    port_b,0   ;
min      nop        ;
movlw    0x20       ;checa se se datao eh zero
subwf    datao,1    ;
btfss    status,0   ;
movwf    datao      ;
call     WRBYTE     ;
movlw    0x20       ;
subwf    datao,0    ;
btfsc    status,2   ;
goto     zerob      ;
movlw    0x20       ;
addwf    datao,1    ;
voltage goto     ajuste ;
sw_memo bsf      port_a,5 ;
call     WAIT1      ;
btfss    port_a,0   ;debouce de sw_memo
goto     sw_memo    ;
movf     slavem,w   ;seta end slave e modo escrit
movwf    slave      ;escrever na memoria
movf     addrm,w    ;
movwf    addr       ;
call     FA1        ;
call     WRBYTE     ;
call     WAIT1      ;
bcf     port_a,5    ;
goto     opcao      ;
zerob    clr     datao ;
goto     voltab     ;
;*****
rcoff_sw movlw    0xdf    ;seta porta b

```

```

    tris    port_b          ;
    bsf     port_c,1       ;acende led ind. rcoff
    movlw   0xa0           ;seta slave como sendo a0h
    movwf   slavem        ;memoria
    movlw   0x19           ;seta end da memo como 19h
    movwf   addrm         ;rcoff memoria
    movlw   0xba          ;seta slave como sendo bah
    movwf   slave         ;vcj (escrita)
    movlw   0x0d          ;seta sub end. como 0dh
    movwf   addr         ;rcoff vcj
    call    FA1           ;
    call    READ          ;chama sub rotina de leitura
    call    WAIT1         ;
    ;bsf    port_b,5      ;
    movf    datai,w       ;copia datai em
    movwf   datao         ;datao
    goto    ajustel      ;

```

```

-----
ajustel
    movlw   b'11011111'   ;seta port_b
    tris    port_b        ;
    btfss   port_b,0     ;testa bit 0 da porta b
    goto    opcao        ;se 0, vai p/opcao
    btfss   port_b,1     ;testa bit 1 da porta b
    goto    opcao        ;se 0 vai p/opcao
    btfss   port_a,0     ;le sw_memo skip se nao zero
    goto    sw_memol     ;se zero vai para sw memo
    btfss   port_a,2     ;le sw_inc skip se nao zero
    goto    sw_incl     ;se zero vai para sw_inc
    btfss   port_a,3     ;le sw_dec skip se nao zero
    goto    sw_decl     ;se zero vai sw_dec
    goto    ajustel      ;
sw_incl   btfsc   port_b,5
    call    FA1          ;
    call    WAIT3        ;
    movlw   0xff         ;maximo 255d para r-cutoff
    subwf   datao,w      ;
    btfsc   status,z    ;
    goto    max1        ;
voltrd1   nop          ;
    incf    datao,1     ;incrementa dado
voltrd2   nop          ;
    call    WRBYTE       ;escreve no ic vcj
    goto    ajustel     ;volta para ajustel
sw_decl   btfsc   port_b,5
    call    FA1          ;
    call    WAIT3        ;
    movlw   0x00         ;
    addwf   datao,w      ;
    btfsc   status,z    ;
    goto    min1        ;
voltrd3   nop          ;
    decf    datao,1     ;
voltrd4   nop          ;
    call    WRBYTE       ;escreve no vcj
    goto    ajustel     ;volta para ajustel
sw_memol  bsf     port_a,5
    call    WAIT1        ;
    btfss   port_a,0     ;debouce de sw_memo
    goto    sw_memol    ;
    movf    slavem,w     ;seta end slave e modo escrit
    movwf   slave       ;escrever na memoria
    movf    addrm,w      ;
    movwf   addr        ;
    call    FA1         ;

```

```

call    WRBYTE      ;
call    WAIT1       ;
movlw   0x1f        ;
movwf   addr        ;
movf    co_sw,0     ;
movwf   datao       ;
call    WRBYTE      ;
call    WAIT1       ;
bcf     port_a,5    ;
goto    opcao       ;
max1    btfsc       co_sw,5      ;testa bit 5 de co_sw
goto    voltrd2     ;se zero, volta
bsf     co_sw,5     ;seta bit 5 de co_sw
movf    datao,w     ;coloca datao no reg w
movwf   tempdt      ;coloca datao em tempdt
movf    co_sw,w     ;pega co_sw, coloca em w
movwf   datao       ;coloca w em datao
movlw   0x10        ;seta end. co_sw no vcj
movwf   addr        ;coloca w em addr
call    WRBYTE      ;escreve no vcj
movlw   0x0d        ;seta end r-cutoff
movwf   addr        ;do vcj
movf    tempdt,w    ;
movwf   datao       ;
goto    voltrd1     ;volta
mini    btfss       co_sw,5      ;testa bit 5 de co_sw
goto    voltrd4     ;se zero, volta
bcf     co_sw,5     ;reseta bit 5 de co_sw
movf    co_sw,0     ;pega co_sw, coloca em w
movwf   datao       ;coloca w em datao
movlw   0x10        ;seta end. co_sw no vcj
movwf   addr        ;coloca w em addr
call    WRBYTE      ;chama rotina de escrita
movlw   0x0d        ;seta end. r-cutoff
movwf   addr        ;do vcj
goto    voltrd3     ;volta
;*****
gcoff_sw      ;
movlw   0xdf        ;seta porta b
tris    port_b      ;
bsf     port_c,2    ;acende led ind. gcoff
movlw   0xa0        ;seta slave como sendo a0h
movwf   slavem      ;memoria
movlw   0x1a        ;seta end da memo como lah
movwf   addrm       ;gcoff memoria
movlw   0xba        ;seta slave como sendo bah
movwf   slave       ;vcj (escrita)
movlw   0x0e        ;seta sub end. como 0eh
movwf   addr        ;gcoff vcj
call    FAl        ;
call    READ        ;chama sub rotina de leitura
call    WAIT1       ;
;bsf     port_b,5    ;
movf    datai,w     ;copia datai em
movwf   datao       ;datao
goto    ajuste2     ;
-----
ajuste2
movlw   b'11011111' ;seta port_b
tris    port_b      ;
btfss   port_b,0    ;testa bit 0 da porta b
goto    opcao       ;se 0, vai p/opcao
btfss   port_b,1    ;testa bit 1 da porta b
goto    opcao       ;se 0 vai p/opcao
btfss   port_a,0    ;le sw_memo skip se nao zero
goto    sw_memo2    ;se zero vai para sw memo

```

```

        btfss    port_a,2      ;le sw_inc skip se nao zero
        goto    sw_inc2      ;se zero vai para sw_inc
        btfss    port_a,3      ;le sw_dec skip se nao zero
        goto    sw_dec2      ;se zero vai sw_dec
        goto    ajuste2      ;ajuste de g_cutoff
sw_inc2 btfsc    port_b,5      ;testa bit 5 porta_b
        call    FA1          ;desab. up e hab. o ucontrolador
        call    WAIT3        ;espere um pouco
        movlw   0xff         ;maximo 255d para g-cuttoff
        subwf   datao,w      ;compara se datao igual ffh
        btfsc   status,z     ;se nao, bit de flag e zero e skip
        goto    max2        ;
voltgn1 nop          ;
        incf    datao,1      ;incrementa dado
voltgn2 nop          ;desab. o uP e habil. o controlador
        call    WRBYTE       ;escreve no ic vcj
        goto    ajuste2      ;
sw_dec2 btfsc    port_b,5      ;
        call    FA1          ;
        call    WAIT3        ;
        movlw   0x00         ;testa se dac esta no minimo
        addwf   datao,w      ;se esta no minimo, nao incre
        btfsc   status,z     ;menta mais
        goto    min2        ;
voltgn3 nop          ;
        decf    datao,1      ;
voltgn4 nop          ;
        call    WRBYTE       ;
        goto    ajuste2      ;
sw_memo2
        bsf    port_a,5      ;
        call    WAIT1        ;
        btfss   port_a,0      ;debouce de sw_memo
        goto    sw_memo2     ;
        movf   slavem,w      ;seta end slave e modo escrit
        movwf  slave        ;escrever na memoria
        movf   addrm,w      ;seta endereco para escrita
        movwf  addr         ;
        call   FA1          ;desab up e hab ucontr.
        call   WRBYTE       ;
        call   WAIT1        ;
        movlw  0x1f         ;
        movwf  addr         ;
        movf   co_sw,0      ;
        movwf  datao        ;
        call   WRBYTE       ;
        call   WAIT1        ;espere um pouco
        bcf    port_a,5      ;
        goto   opcao        ;
max2    btfsc   co_sw,6      ;testa bit 6 de co_sw
        goto   voltgn2     ;se zero, volta
        bsf    co_sw,6      ;seta bit 5 de co_sw
        movf   datao,w      ;
        movwf  tempdt      ;
        movf   co_sw,w      ;pega co_sw, coloca em w
        movwf  datao        ;coloca w em datao
        movlw  0x10         ;seta end. co_sw no vcj
        movwf  addr         ;coloca w em addr
        call   WRBYTE       ;escreve no vcj
        movlw  0x0e         ;seta end g-cuttoff
        movwf  addr         ;do vcj
        goto   voltgn1     ;volta
min2    btfss   co_sw,6      ;testa bit 6 de co_sw
        goto   voltgn4     ;se zero, volta
        bcf    co_sw,6      ;reseta bit 6 de co_sw
        movf   co_sw,0      ;pega co_sw, coloca em w
        movwf  datao        ;coloca w em datao

```

```

movlw 0x10          ;seta end. co_sw no vcj
movwf addr         ;coloca w em addr
call  WRBYTE       ;chama rotina de escrita
movlw 0x0e         ;seta end. g-cutoff
movwf addr         ;do vcj
goto  voltgn3      ;volta
;*****
bcoff_sw
movlw 0xdf         ;seta porta b
tris  port_b       ;
bsf   port_c,3     ;acende led ind. bcoff
movlw 0xa0         ;seta slave como sendo a0h
movwf slavem       ;memoria
movlw 0x1b         ;seta end da memo como 1bh
movwf addrm        ;bcoff memoria
movlw 0xba         ;seta slave como sendo bah
movwf slave        ;vcj (escrita)
movlw 0x0f         ;seta sub end. como 07h
movwf addr         ;bcoff vcj
call  FA1          ;
call  READ         ;chama sub rotina de leitura
call  WAIT1        ;
;bsf   port_b,5    ;
movf  data1,w      ;copia data1 em
movwf datao        ;datao
goto  ajuste3      ;
-----
ajuste3
movlw b'11011111' ;seta port_b
tris  port_b       ;
btfss port_b,0     ;testa bit 0 da porta b
goto  opcao        ;se 0, vai p/opcao
btfss port_b,1     ;testa bit 1 da porta b
goto  opcao        ;se 0 vai p/opcao
btfss port_a,0     ;le sw_memo skip se nao zero
goto  sw_memo3     ;se zero vai para sw_memo
btfss port_a,2     ;le sw_inc skip se nao zero
goto  sw_inc3      ;se zero vai para sw_inc
btfss port_a,3     ;le sw_dec skip se nao zero
goto  sw_dec3      ;se zero vai sw_dec
goto  ajuste3      ;
sw_inc3
btfsc port_b,5     ;
call  FA1          ;
call  WAIT1        ;
movlw 0xff         ;maximo 255d para b-cutoff
subwf datao,w      ;
btfsc status,z     ;
goto  max3         ;incf datao,1
voltbe1 nop        ;
incf  datao,1      ;incrementa dado
voltbe2 nop        ;desab. o uP e habil. o control.
call  WRBYTE       ;escreve no ic vcj
goto  ajuste3      ;
sw_dec3
btfsc port_b,5     ;
call  FA1          ;
call  WAIT1        ;
movlw 0x00         ;
addwf datao,w      ;
btfsc status,z     ;
goto  min3         ;
voltbe3 nop        ;
decf  datao,1      ;
btfsc port_b,0     ;
voltbe4 nop        ;
call  WRBYTE       ;

```

```

goto    ajuste3      ;
sw_memo3
bsf     port_a,5     ;
call    WAIT1        ;
btfss   port_a,0     ;debouce de sw_memo
goto    sw_memo3     ;
movf    slavem,w     ;seta end slave e modo escrit
movwf   slave        ;escrever na memoria
movf    addrm,w      ;
movwf   addr         ;
call    FA1          ;
call    WRBYTE       ;
call    WAIT1        ;
movlw   0x1f         ;
movwf   addr         ;
movf    co_sw,0      ;
movwf   datao        ;
call    WRBYTE       ;
call    WAIT1        ;
bcf     port_a,5     ;
goto    opcao        ;
max3
btfsc   co_sw,7      ;testa bit 7 de co_sw
goto    voltbe2      ;se zero, volta
bsf     co_sw,7      ;seta bit 7 de co_sw
movf    datao,w      ;pega datao e joga em w
movwf   tempdt       ;coloca datao em tempdt
movf    co_sw,w      ;pega co_sw, coloca em w
movwf   datao        ;coloca w em datao
movlw   0x10         ;seta end. co_sw no vcj
movwf   addr         ;coloca w em addr
call    WRBYTE       ;escreve no vcj
movlw   0x0f         ;seta end b-cutoff
movwf   addr         ;do vcj
goto    voltbe1      ;volta
min3
btfss   co_sw,7      ;testa bit 7 de co_sw
goto    voltbe3      ;se zero, volta
bcf     co_sw,7      ;reseta bit 7 de co_sw
movf    co_sw,w      ;pega co_sw, coloca em w
movwf   datao        ;coloca w em datao
movlw   0x10         ;seta end. co_sw no vcj
movwf   addr         ;coloca w em addr
call    WRBYTE       ;chama rotina de escrita
movlw   0x0f         ;seta end. b-cutoff
movwf   addr         ;do vcj
goto    voltbe4      ;volta
;*****
;          ROTINA PARA RESETAR FA1
FA1      bcf     port_b,5     ;
call    WAIT2        ;
nop     ;
nop     ;
nop     ;
nop     ;
nop     ;
nop     ;
retlw   0
;*****
;          ROTINA PARA RESETAR FA1
FA12     bcf     port_b,5     ;
;call   WAIT1        ;
nop     ;
nop     ;
nop     ;
nop     ;
nop     ;
nop     ;

```

```

nop
retlw 0
;*****
;
;          rotina de atraso WAIT
;esta rotina pega o valor em loops e multiplica por lms para
;determinar o tempo de atraso.
;*****

WAIT
top      movlw   .110           ;variavel de ajuste de tempo
        movwf   loops2        ;
top2     nop           ;sente e espere
        nop           ;
        nop           ;
        nop           ;
        nop           ;
        decfsz  loops2        ;loop interno completo?
        goto   top2          ;nao, va novamente
        decfsz  loops        ;loop externo completo?
        goto   top           ;nao, va novamente
        retlw   0            ;
;*****
;
;          rotina de atraso WAIT1
;esta rotina pega o valor em loops e multiplica por lms para
;determinar o tempo de atraso.
;*****

WAIT1
top1     movlw   .30           ;variavel de ajuste de tempo
        movwf   loops2        ;
top3     nop           ;sente e espere
        nop           ;
        nop           ;
        nop           ;
        nop           ;
        decfsz  loops2        ;loop interno completo?
        goto   top3          ;nao, va novamente
        decfsz  loops        ;loop externo completo?
        goto   top1          ;nao, va novamente
        retlw   0            ;
;*****
;
;          rotina de atraso WAIT2
;esta rotina pega o valor em loops e multiplica por lms para
;determinar o tempo de atraso.
;*****

WAIT2
top4     movlw   .97           ;variavel de ajuste de tempo
        movwf   loops2        ;
top5     nop           ;sente e espere
        nop           ;
        nop           ;
        nop           ;
        nop           ;
        decfsz  loops2        ;loop interno completo?
        goto   top5          ;nao, va novamente
        decfsz  loops        ;loop externo completo?
        goto   top4          ;nao, va novamente
        retlw   0            ;
;*****
;
;          rotina de atraso WAIT3
;esta rotina pega o valor em loops e multiplica por lms para
;determinar o tempo de atraso.

```

```

;*****
WAIT3
top6   movlw   .15           ;variavel de ajuste de tempo
      movwf   loops2        ;
top7   nop           ;sente e espere
      nop           ;
      nop           ;
      nop           ;
      nop           ;
      decfsz  loops2        ;loop interno completo?
      goto   top7          ;nao, va novamente
      decfsz  loops        ;loop externo completo?
      goto   top6          ;nao, va novamente
      retlw  0             ;
;*****
;          sub rotina bit de partida
;esta subrotina gera o bit de partida
;(linha de dados vai para L enquanto clock esta H)
;*****
;
BSTART bsf      port_b,sdata   ;assegura linha de dados H
      movlw   b'00011111'    ;seta linhas de dados e clock
      tris   port_b          ;como saida
      bcf    port_b,sclk     ;assegura clock L
      nop           ;
      bsf    port_b,sclk     ;seta clock H
      nop           ;
      nop           ;
      nop           ;
      nop           ;
      nop           ;
      bcf    port_b,sdata    ;linha de dados vai para L
      nop           ;enquanto clock esta H
      nop           ;(bit de partida)
      nop           ;temporizacao
      nop           ;
      nop           ;
      bcf    port_b,sclk     ;trem de clock partida
      nop           ;
      nop           ;
      retlw  0             ;
;*****
;          subrotina bit de parada
;esta rotina gera um bit de parada
;(linha de dados vai para H enquanto clock esta H)
;*****
;
BSTOP  movlw   b'00011111'    ;seta linhas de dados e clock
      tris   port_b          ;como saidas
      bcf    port_b,sdata    ;assegura linha de dados L
      nop           ;
      nop           ;
      nop           ;
      bsf    port_b,sclk     ;seta clock H
      nop           ;
      nop           ;
      nop           ;
      bsf    port_b,sdata    ;linha de dados vai para H
      nop           ;enquanto clock esta H
      nop           ;
      bcf    port_b,sclk     ;seta clock L novamente
      nop           ;
      nop           ;
      nop           ;
      retlw  0             ;
;*****

```



```

;          subrotina bitout
;esta subrotina pega um bit de dados em 'do' e transmite para
;a memoria.
;*****
;
BITOUT  movlw   b'00011111'      ; seta dados e clock como
      tris    port_b            ; saidas
      btfs   eeprom,do         ; checa estado do bit de dados
      goto   bitlow            ; para transmissao
      bsf    port_b,sdata      ; bit H, seta linha de dados H
      goto   clkout            ; go toggle the clock
bitlow  bcf    port_b,sdata     ; saida e um bit L
clkout  bsf    port_b,sclk     ; seta linha de clock H
      nop    ;
      nop    ;
      nop    ;
      nop    ;
      bcf    port_b,sclk       ; retorna clock para L
      retlw  0                 ;
;*****
;          subrotina bitin (escrita)
;le um bit de dados na memoria e o armazena em 'di'
;*****
;
BITINW  bsf    eeprom,di        ; assume que o bit de ent. e H
      movlw  b'10011111'      ; faz sdata como entrada
      tris   port_b            ;
      bsf   port_b,sdata      ; seta sdata como entrada
      bsf   port_b,sclk       ; seta clock H
      nop   ;
      nop   ;
      nop   ;
      nop   ;
      btfs  port_b,sdata      ; le o bit de dado
      bcf   eeprom,di         ; bit de entrada eh L
      bcf   port_b,sclk       ; seta clock baixo novamente
      retlw 0                 ;
;*****
;          subrotina bitin (leitura)
;le um bit de dados na memoria e o armazena em 'di'
;*****
;
BITINR  bsf    eeprom,di        ; assume que o bit de ent. e H
      movlw  b'10011111'      ; faz sdata como entrada
      tris   port_b            ;
      bsf   port_b,sclk       ; seta clock H
      nop   ;
      nop   ;
      nop   ;
      nop   ;
      btfs  port_b,sdata      ; le o bit de dado
      bcf   eeprom,di         ; bit de entrada eh L
      bcf   port_b,sclk       ; seta clock baixo novamente
      retlw 0                 ;
;*****
;          subrotina de transmissao de dados (escrita)
;esta rotina pega um byte de dados armazenado em 'datao' e
;transmite para a memoria eeprom. Sera enviado um clock a mais
;para reconhecimento. Se o bit ack da memoria for L, entao a
;transmissao foi bem sucedida. se o bit ack for H, entao a me-
;a memoria nao enviou o bit ack e o led ack fail acendera.
;*****
;
TXW     movlw  .8                ; seta numero de bits para 8

```

```

movwf    count                ;
TXLPR    bcf    eeprom,do      ;assume bit de saida eh L
          btfsc txbuf,7        ;o bit eh baixo mesmo?
          bsf    eeprom,do      ;senao, bit eh H
          call   BITOUT        ;chama SR bitout
          rlf   txbuf          ;roda txbuf para esquerda
          decfsz count         ;8 bits transmitidos?
          goto  TXLPR         ;nao, va novamente
          call   BITINW       ;le bit de reconhecimento
          btfsc eeprom,di      ;checa bit de reconhecimento
          bsf    port_a,ackf    ;falha de reconhecimento LED ON
          retlw  0             ;

```

```

;*****
;
;      subrotina de transmissao de dados (leitua)
;esta rotina pega um byte de dados armazenado em 'datao' e
;transmite para a memoria eeprom. Sera enviado um clock a mais
;para reconhecimento. Se o bit ack da memoria for L, entao a
;transmissao foi bem sucedida. se o bit ack for H, entao a me-
;a memoria nao enviou o bit ack e o led ack fail acendera.
;*****

```

```

TXR      movlw   .8            ;seta numero de bits para 8
          movwf  count        ;
TXLPR    bcf    eeprom,do      ;assume bit de saida eh L
          btfsc txbuf,7        ;o bit eh baixo mesmo?
          bsf    eeprom,do      ;senao, bit eh H
          call   BITOUT        ;chama SR bitout
          rlf   txbuf          ;roda txbuf para esquerda
          decfsz count         ;8 bits transmitidos?
          goto  TXLPR         ;nao, va novamente
          call   BITINR       ;le bit de reconhecimento
          btfsc eeprom,di      ;checa bit de reconhecimento
          bsf    port_a,ackf    ;falha de reconhecimento LED ON
          retlw  0             ;

```

```

;*****
;
;      Sub-rotina para recepcao de dados
;Esta sub rotina le um byte de dados de um componete e coloca
;no registro datai e entao envia um ack high para indicar que
;nao ha mais dados para serem lidos

```

```

RX
          movlw   .8            ;seta bits em 8
          movwf  count        ;
          clrf   datai         ;
RXLP     bcf    status,0      ;
          rlf   datai         ;
          call   BITINR       ;
          btfsc eeprom,di      ;
          bsf    datai,0      ;
          decfsz count         ;8 bits recebidos?
          goto  RXLP         ;
          retlw  0             ;

```

```

;*****
;
;      rotina byte write
;esta rotina escreve o dado em 'datao' em oito enderecos con-
;secutivos da memoria a partir do endereco 00. Esta rotina
;espera 10ms para dar tempo de a escrita ser feita. este pro-
;grama se repete.
;*****

```

```

WRBYTE   bcf    port_a,ackf    ;apaga led ack ng
byte     call   BSTART        ;gera bit de partida
          movf  slave,w        ;pega end. do slave e coloca
          movwf txbuf         ;no buffer de transmissao
          call   TXW          ;e o transmite
          movf  addr,w        ;pega endereco da memo , coloca no
          movwf txbuf         ;buffer de transmissao

```

```

    call    TXW          ;e o transmite
    movf   datao,w      ;pega o byte de dados, coloca no
    movwf  txbuf        ;buffer de transmissao
    call   TXW          ;e o transmite
    call   BSTOP        ;gera bit de parada
    retlw  0
;*****
;
;          READ
;Esta sub rotina le o conteudo do endereco especificado no
;slave especificado.
;*****
READ
    bcf    port_a,ackf  ;apaga led ack fail se aceso
    call   BSTART      ;gera bit de partida
    movf   slavem,w    ;pega end. do slave
    movwf  txbuf       ;coloca no buffer de transm.
    call   TXR         ;e envia
    movf   addrm,w     ;pega o endereco
    movwf  txbuf       ;coloca no buffer de transm.
    call   TXR         ;e o envia
    call   BSTART      ;gera bit de partida
    movlw  b'10100001' ;end. slave e modo leitura
    movwf  txbuf       ;coloca no buffer de transm.
    call   TXR         ;e o transmite
rbyte    call   RX      ;chama sub rotina RX
    bsf    eeprom,do   ;sim, envia ack H.
    call   BITOUT      ;
    call   BSTOP      ;
    retlw  0
;*****
temp     call   WAIT    ;///
    decfsz tempstr    ;///
    goto   temp        ;///
    return 0           ;///
end

```