

Vamos debater a função de escrita no display, ai passamos para a próxima função que será a de teclado.

Declarações

```
/*-----DECLARACAO DAS BIBLIOTECAS-CONSTANTES E PINOS DE CONTROLE-----*/  
  
#include "AT89S8252.h" /*Declara a biblioteca para AT89S8252 ATMEL modificada por CARLOS R. LAURINDO */  
#define displayporta P0 /*Seleciona a porta que escreve a palavra de 8 bits no display */  
#define RS P2_6 /*Pino que abilita o display */  
#define E P2_7 /*Gera strobe no display */
```

O importante do código acima são as linhas define, displayporta note que nesta linha e que definimos a porta do microcontrolador onde o display esta conectado fisicamente neste caso a porta 0 ou seja (#define displayporta P0), caso o display estivesse conectado a porta 1 ficaria (#define displayporta P1) e assim por diante para as demais portas. Os pinos RS e E são para uso do hardware do display enable E e Estrobe, e são definidos através dos (#define RS P2_6) designa um pino do microcontrolador fisicamente ou seja o BIT6 da porta P2 para RS. Já para (#define E P2_7) designa um pino do microcontrolador fisicamente, ou seja, BIT7 da porta P2 para E.

Funções de baixo nível para escrita no display.

Todo o código abaixo e responsável por implantar apenas três funções as quais darão total controle sobre o display. Posteriormente este código será incluso na biblioteca padrão AT898252.h apenas as funções linha(), strw(), escreve() serão utilizadas diretamente na programação. Por ser razoavelmente, complicado e inútil descrevê-lo. As funções implantadas serão discutidas mais a baixo.

Código que implanta as funções linha(), strw() e escreve().

```
/*-----ROTINAS DE CONTROLE DO DISPLAY TODAS AS ROTINAS ABAIXO SAO UTILIZADAS NO CONTROLE DO DISPLAY-----*/  
  
/* Rotina para escrita no display de cristal liquido escreve strings */  
void cmld(unsigned char c, char cd){int t; displayporta=c; if (cd==0) RS=0; else RS=1; E=0; E=1; for (t = 0; t<8; t++); if ((cd==0) && (c<127)) for (t=0; t<180; t++);}  
  
/* Rotina de inicializacao do display */  
void initld(void){ cmld(0x30,0); cmld(0x30,0); cmld(0x30,0); cmld(0x38,0); cmld(0x0c,0); cmld(0x01,0); cmld(0x06,0);}  
  
/* Rotina que escreve uma cadeia de caracteres no display */  
void strw(char *c){ for (; *c!=0; c++) cmld(*c,1);}  
  
/* Rotina que posiciona o cursor na linha selecionada linha(1) ou linha(2) primeiro caracter da linha selecionada deixando pronto para escrever no display */  
void linha(char ln) { if (ln==1){ cmld(0x80,0);}; if (ln==2) { cmld(0xc0,0);}  
  
/* Rotina de temporizacao */  
void espera(unsigned int quanto) {register int t;t=0; do { t=t+1; } while (t<quanto);}  
  
/* Rotina de escrita no display de variaveis */  
void wrld(unsigned char c){ char t; displayporta=c; RS=1; E=0; E=1; for (t = 0; t < 10; t++);}  
  
/* Rotina que imprime o numero de dígitos de um inteiro no display utilizada para visualizar variaveis no display */  
void escreve(unsigned long int i, unsigned char digitos, unsigned char cortar)  
{  
    if ((digitos>9)&&(cortar<=9)){ wrld(0x30 | ((i/ 1000000000)% 10));}  
    if ((digitos>8)&&(cortar<=8)){ wrld(0x30 | ((i/ 100000000)% 10));}  
    if ((digitos>7)&&(cortar<=7)){ wrld(0x30 | ((i/ 10000000)% 10));}  
    if ((digitos>6)&&(cortar<=6)){ wrld(0x30 | ((i/ 1000000)% 10));}  
    if ((digitos>5)&&(cortar<=5)){ wrld(0x30 | ((i/ 100000)% 10));}  
    if ((digitos>4)&&(cortar<=4)){ wrld(0x30 | ((i/ 10000)% 10));}  
    if ((digitos>3)&&(cortar<=3)){ wrld(0x30 | ((i/ 1000)% 10));}  
    if ((digitos>2)&&(cortar<=2)){ wrld(0x30 | ((i/ 100)% 10));}  
    if ((digitos>1)&&(cortar<=1)){ wrld(0x30 | ((i/ 10)% 10));}  
    if ((digitos>0)&&(cortar<=0)){ wrld(0x30 | ((i/ 1)% 10));}  
}  
  
/*-----Final do bloco de rotinas especificas do diplay-----*/
```

Apenas as três funções descritas abaixo serão efetivamente utilizadas durante a programação, para total controle do display de forma fácil e intuitiva o código acima apenas implanta as funções e não será descrito.

São três as funções:

01-) Função - `linha(1);` Esta função é responsável por posicionar o cursor em uma linha do display em específico por exemplo se você quiser colocar o cursor na linha 1 escreva – `linha(1)`, caso queira posicionar o cursor na linha 2 do display escreva – `linha(2)`, esta função posiciona o cursor na linha especificada no primeiro caractere a esquerda. Deixando o display pronto para a execução das próximas duas funções, `strw()` ou `escreve(,0,0)`.

01-) Função - `strw("");` Esta função escreve strings no display sendo uma verdadeira panacéia escrevendo a partir da posição do cursor qualquer caractere ASCII colocado dentro das aspas duplas Exemplo – `strw("Estou vivo");` escreve no display Estou Vivo a partir de onde o cursor estiver posicionado então o código a seguir:

```
linha(1);  
strw("Estou vivo");
```

Escreve na linha 1 do display Estou vivo a função `strw()` posiciona automaticamente o cursor no último caractere escrito então você pode continuar escrevendo sem se preocupar em posicionar o cursor novamente. Ex. para um display de 16 caracteres por linha:

```
linha(1);  
strw("Estou vivo"); escreve Estou vivo e posiciona automaticamente o cursor no caractere  
10 sobrando ainda 6 caracteres da primeira linha do display podemos continuar assim.  
strw(" Viva!");
```

Resultando no display linha 1 (`Estou vivo Viva!`) assim utilizando os 16 caracteres da primeira linha do display. Fácil não.

Caso queiramos apagar a linha 1 basta escrever:

```
linha(1);  
strw(" "); pronto a linha 1 esta em branco.
```

Para linha 2 tudo igual mudando apenas de `linha(1);` para `linha(2);`

02-) Função - `escreve(,0,0);` Esta função escreve o valor de variáveis no display sendo um pouco mais complexa vamos ver um exemplo:

Caso queiramos escrever o valor de uma variável a qual, por exemplo, tivesse o valor da temperatura ambiente 27.3 graus centígrados, variável chamada temperatura.

`temperatura = 273` note que não existe ponto decimal em programação de baixo nível, então.

`linha(1);` seleciona a linha 1 do display e posiciona o cursor a esquerda

`strw("TMP = ");` escreve uma string no display e posiciona o cursor no caractere 6.

`escreve(temperatura,3,1);` seleciona três dígitos da variável temperatura cortando o primeiro dígito menos significativo

`strw(".");` escreve o ponto decimal para visualização em formato decimal

`escreve(temperatura,1,0);` seleciona um dígito da variável temperatura e não corta nenhum dígito.

`strw(" ");` preenche os demais caracteres da linha 1 com brancos

Resultando no display na linha 1 (`TMP= 27.3`)

Ou seja, a função `escreve(variavel,0,0)` funciona assim `escreve(nome da variável,numero de dígitos,numero de dígitos a cortar da direita para esquerda)`