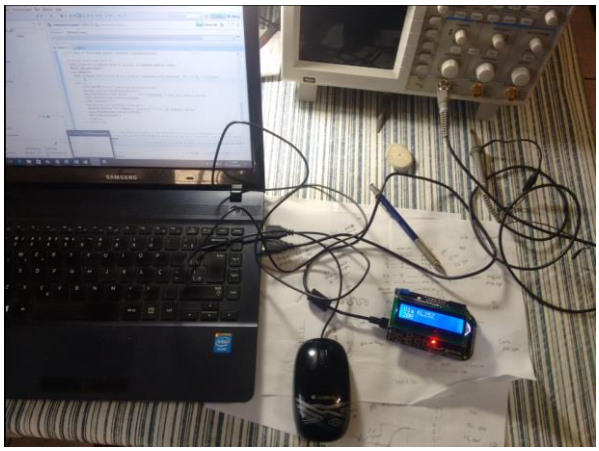
Usando o LCD Shield Arduino no kit FRDM-KI25Z da Freescale



Por Eng. Roberto Bairros dos Santos

www.bairrospd.com

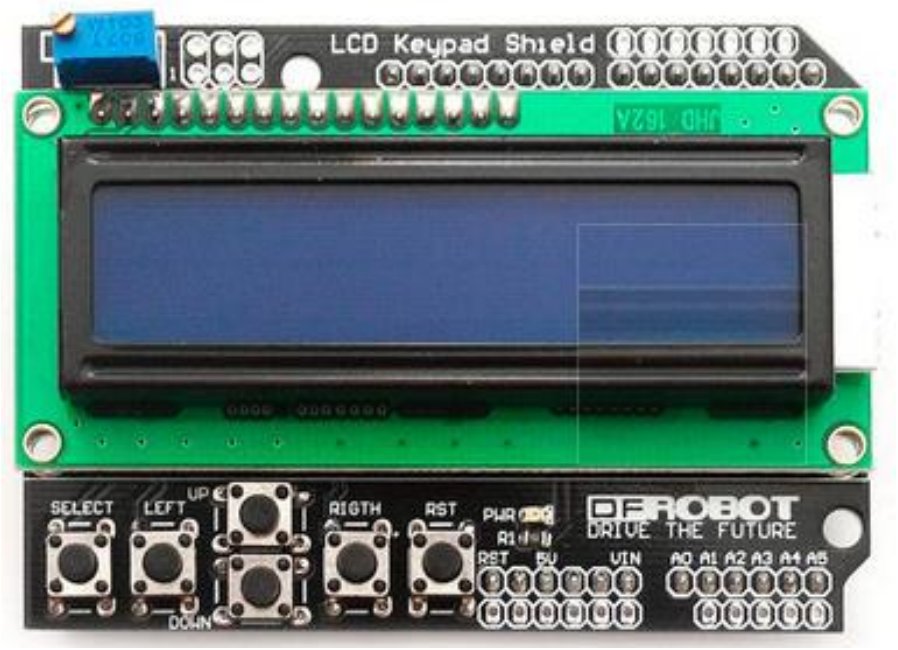
Data: 15/10/2016

Sumário

Introdução.

Este tutorial irá amostrar como usar o shield com LCD e teclado mais comum para a linha Arduino montado no KIT FRDM-KL25Z da Freescale.

O KIT FRDM-KL25Z da Freescale é um kit didático similar ao Arduino, possui um hardware compatível com a placa Arduino UNO de forma a permitir usar os kits, sensores e tudo mais que existe da linha Arduino.



A vantagem do KIT FRDM-KL25Z da Freescale é possuir um microcontrolador MKL25Z128VLK4 de 32 bits ARM CORTEX-MO com 128K de memória Flash e 16KSRAM, com ADC de 16 bits e 53 pinos de IO digital.

A placa do kit já vem com acelerômetro, três leds coloridos, um sensor de toque que simula um potenciômetro.

Você pode conferir os detalhes da placa no link www.freescale.com/FRDM-KL25Z.

O kit já vem com o cabo de programação, como o Arduino.

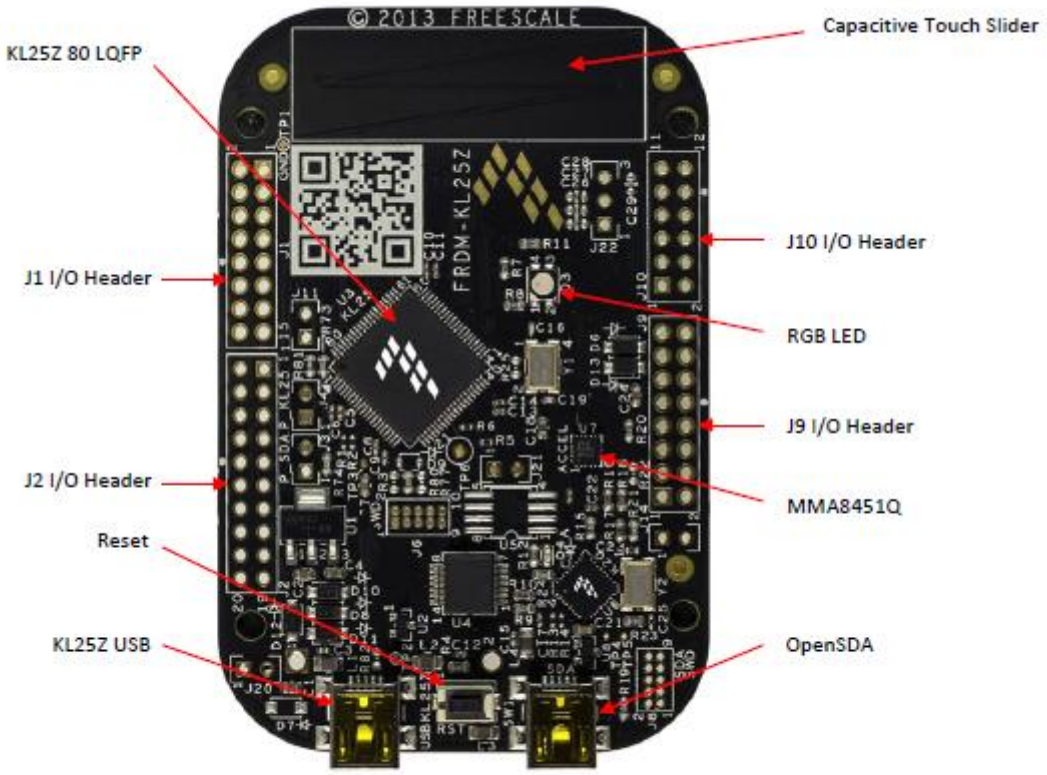
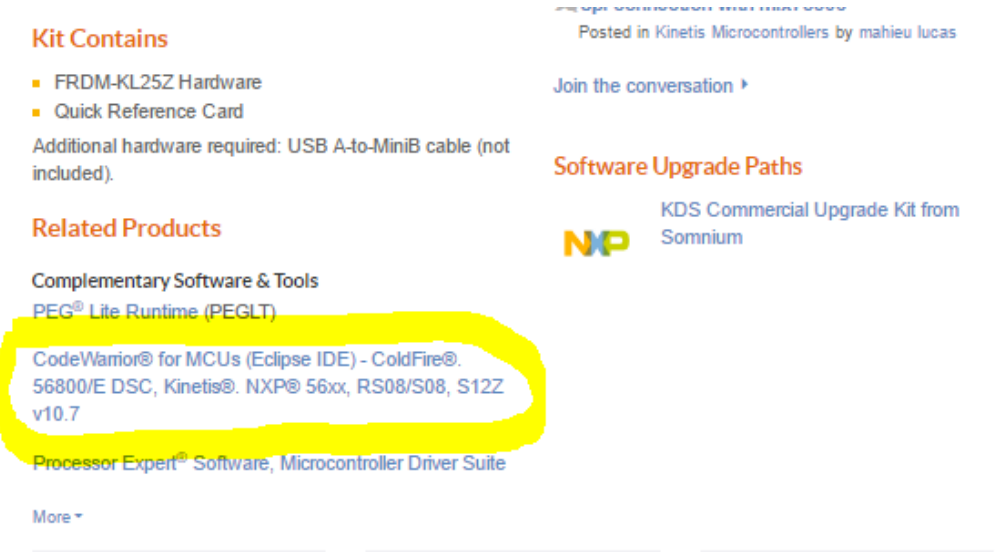


Figure 2. FRDM-KL25Z Feature Call-outs

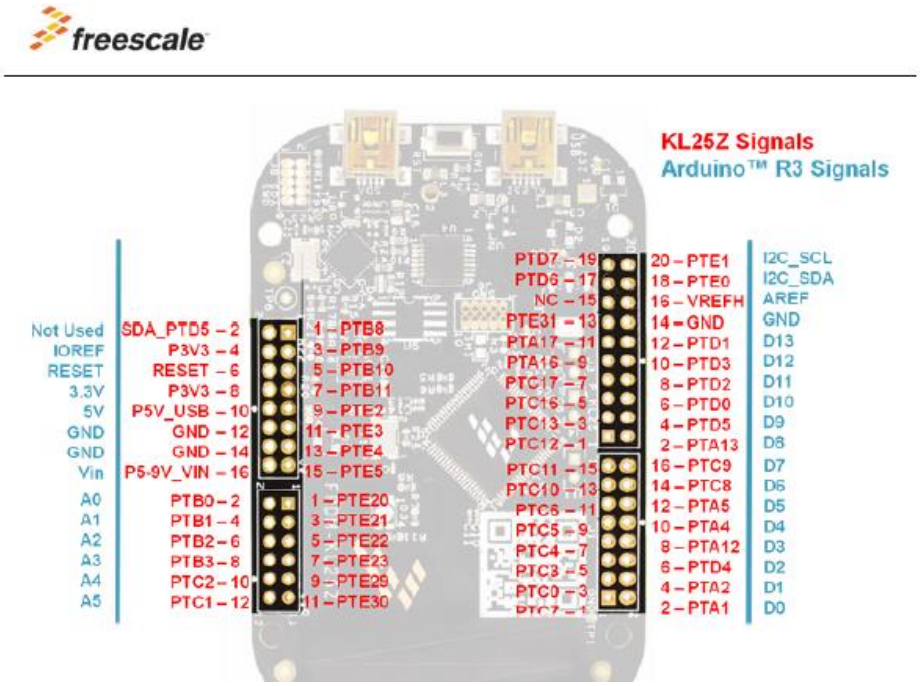
A programação é feita via software chamado CodeWarrior baixado grátis no link dado anteriormente.



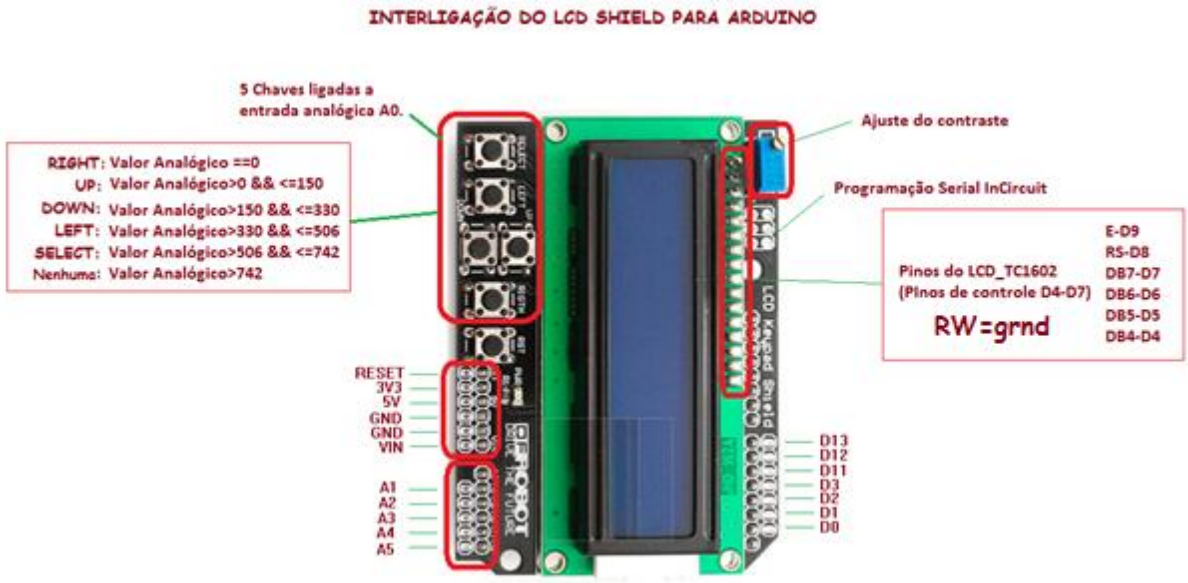
Compatibilidade com a placa UNO do Arduino.

O KIT apresenta um hardware compatível com a placa Arduino UNO de forma que você pode encaixar todos os shields do Arduino diretamente na placa do KIT.

A pinagem é mostrada na figura a seguir.



A interligação do shield é mostrada a seguir.



A Relação entre os endereços dos pinos do LCD e os pinos do kit é mostrada na figura a seguir.

Interligação:

LCD	Arduino	KL25z
E	D9	PTD5
RS	D8	PTA13
DB4	D4	PTA4
DB5	D5	PTA5
DB6	D6	PTC8
DB7	D7	PTC9

O componente LCDHTA

Este trabalho foi baseado na publicação de Erik Stiger no link a seguir.

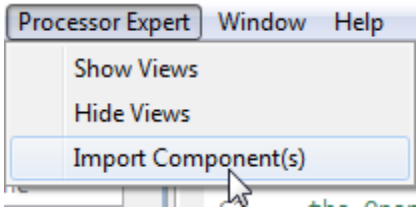
https://mcuoneclipse.com/2012/12/22/hd44780-2x16-character-display-for-kinetis-and-freedom-board/

Para usar o LCD com o CodeWarrior é preciso usar dois componentes criados pelo Erik Stiger o LCDHTA e o WAIT disponível no link descrito a seguir.

http://steinerberg.com/EmbeddedComponents/LCDHTA/home.htm

http://steinerberg.com/EmbeddedComponents/Wait/home.htm

Faça o download dos dois arquivos *.PEupd Importe para o CodeWarrior usando o menu descrito na figura a seguir.

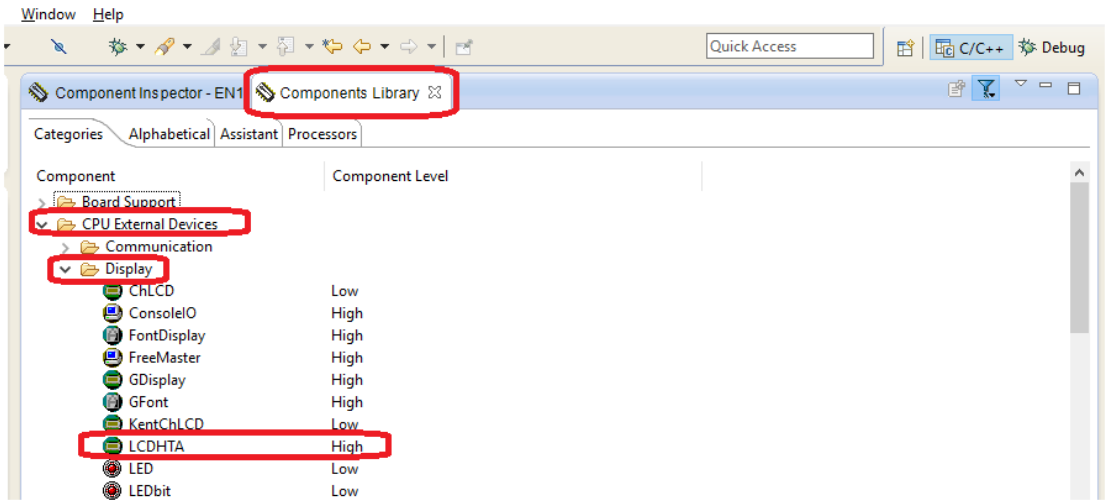


Você pode ver o post do link a seguir que mostra detalhes de como baixar toda a biblioteca criada pelo senhor Erik Stiger.

http://mcuoneclipse.com/2013/05/09/processor-expert-component-peupd-files-on-github/

Você pode pegar este componente na aba da livraria como indica a figura a seguir.

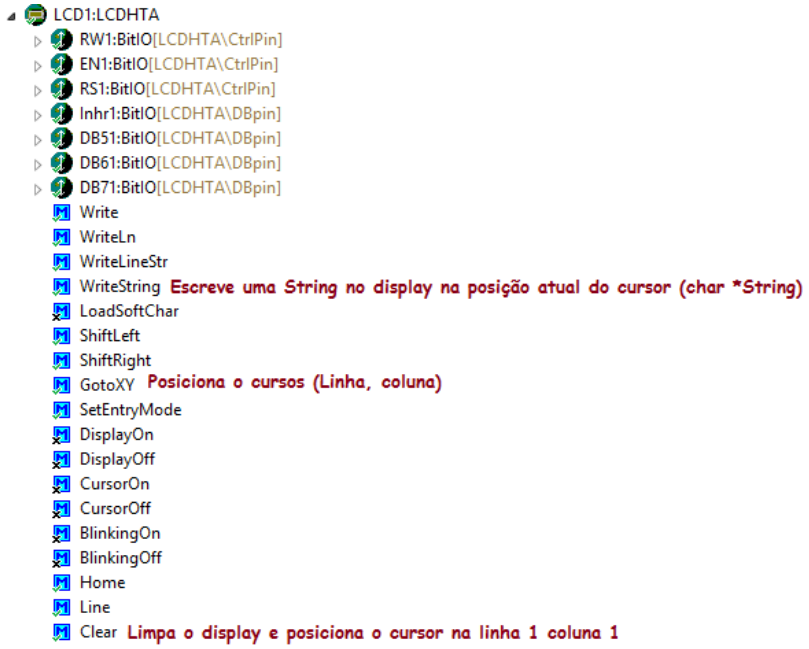
Note que ao carregar o LCDHTA o componente WAIT será carregado também, este é um componente que gera atraso de tempo em ms equivale ao delay(ms) do Arduino!



Como configurar o componente LCDHTA.

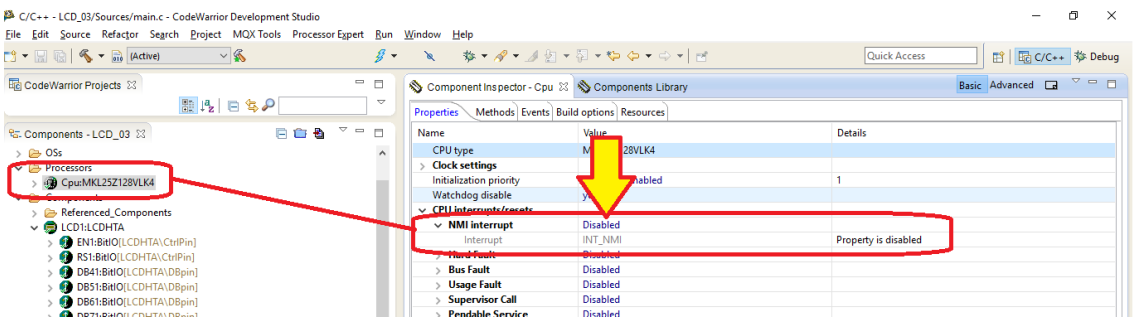
O senhor Erik Stiger diz que 'HTA' é a abreviatura da Universidade de Lucerne onde ele leciona!

O software oferece os métodos descritos abaixo onde estão assinalados os métodos que serão usados neste tutorial.

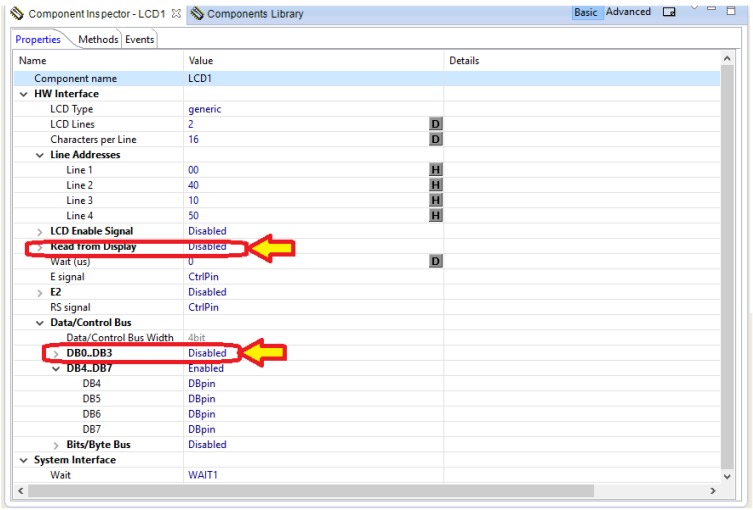


Antes de configurar os pinos você deverá alterar uma configuração da CPU pois o pino PTA4 também usada como interrupção NMI!

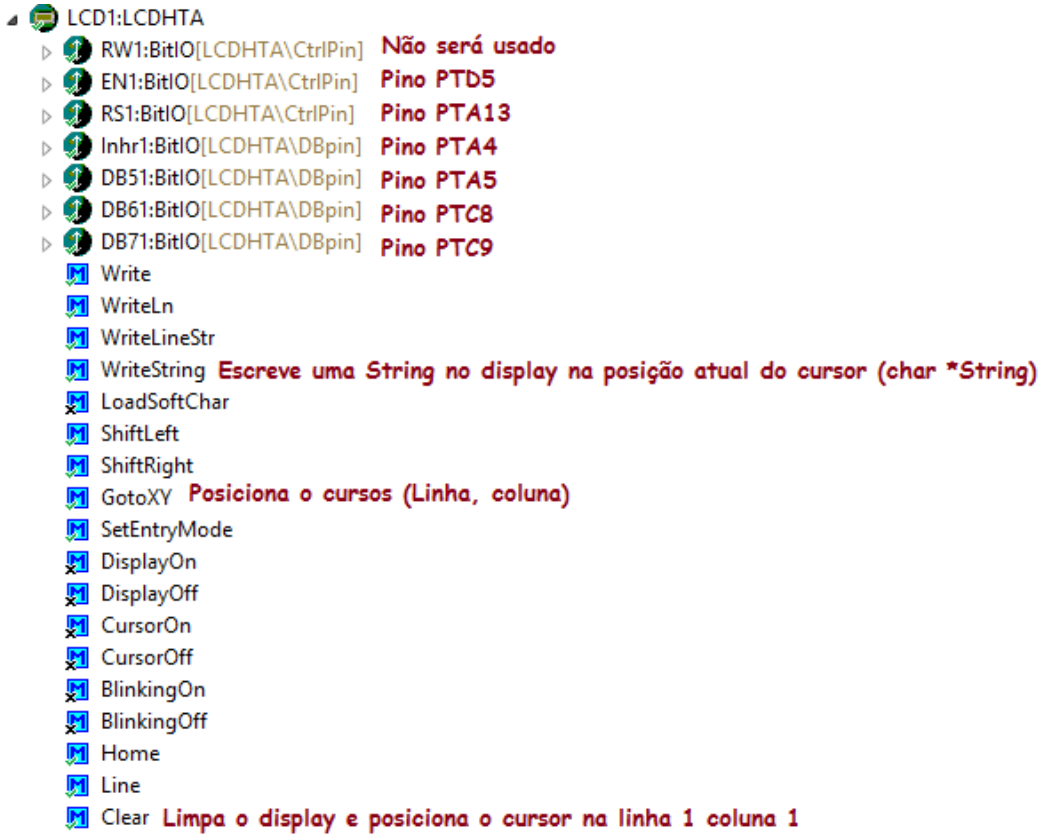
Vá na aba do PE Cpu altere a configuração NMI desabilitando-a, como mostra a figura a seguir.



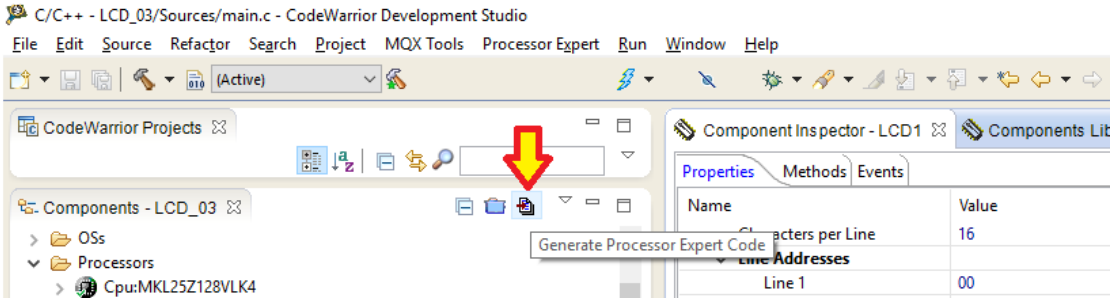
O próximo passo é configurar as propriedades do componente LCD1:LCDHTA para operar com 4 bits sem usar o R/W para isto desabilite o campo Ready from Display e o campo Dat/Control Bus opção DB0...DB3 já que somente os bits DB4 a DB7 serão usados, o restante não precisa alterar.



Agora você já pode alterar os pinos das I/O do componente LCD1:LCDHTA como descrito abaixo, estes são os pinos da placa compatíveis com o shield do LCD Arduino!

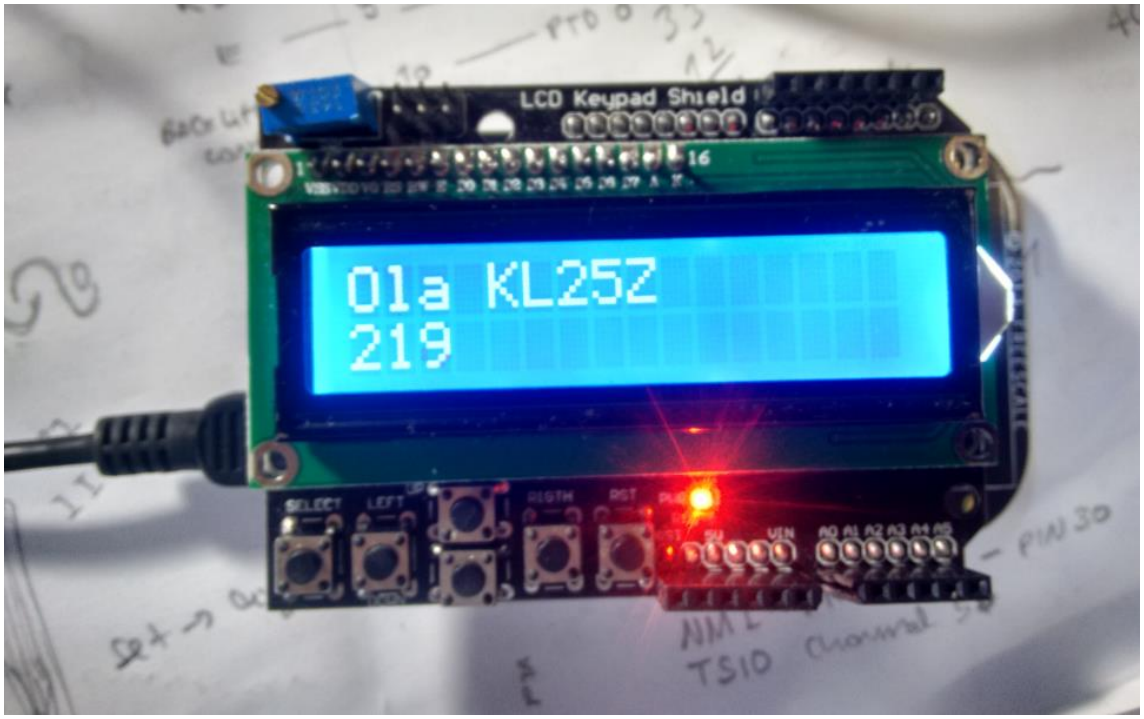


Com o componente configurado não esqueça de gerar o código do PE, depois parta para o programa exemplo.

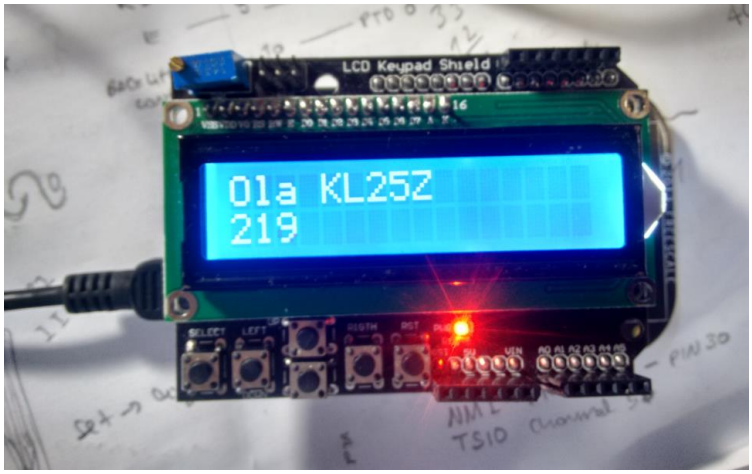


A programação.

O programa irá mostrar na primeira linha do LCD a mensagem “Ola KL25Z” e na segunda linha irá mostrar o tempo em segundos!



Antes do programa contido no laço infinito você deve limpar o display com a instrução LCD1_Clear(), esta função gasta algum tempo executando esta tarefa, por isto, uma função WAIT(1000) de 1000ms deverá ser colocada logo depois do Clear(), por este mesmo motivo o CLEAR deve ser evitado no programa.



Para montar a mensagem com letras e valores numéricos eu prefiro usar a função padrão da linguem “C” sprintf(destivo,string,variáveis) e depois usar a função WriteString(char *String) do componente LCDHTApara escrever no display, onde a String é a mesma variável destino da função sprintf.

Eu também preenchi os espaços em branco para evitar que apareça caracteres indesejáveis na mensagem para isto montei o padrão no formato de comentário para orientar a posição das letras. O exemplo abaixo mostra como escrevo a primeira linha.

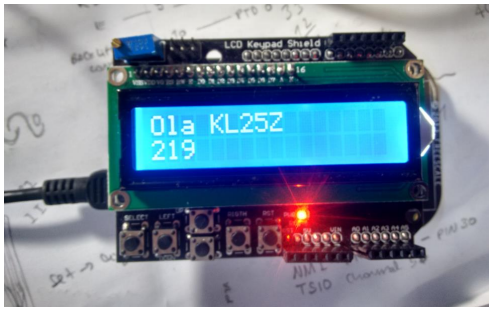
```
char strlcd[17];//string a ser usada no programa oara escrever  no LCD de 16 colunas.

  for(;;) {
      LCD1_GotoXY(1,1);//posiciona primeira linha
    //strcpy(strlcd,"abcdefghijklmnop");
      sprintf(strlcd,"Ola KL25Z       ");//completar 16 col com espaços vazios
      LCD1_WriteString((char*)strlcd);
```

O programa completo é descrito abaixo, você pode baixar o pdf no site www.bairrospd.com copiar e colar o código.

```
/* Write your code here */
 LCD1_Clear();//limpara tudo no inicio, a limpeza demora tempo
 WAIT1_Waitms(1000);
 int tempo=0;
  char strlcd[17];//string a ser usada no programa oara escrever  no LCD de 16 colunas.


       for(;;) {
              LCD1_GotoXY(1,1);//posiciona primeira linha
         //strcpy(strlcd,"abcdefghijklmnop");
              sprintf(strlcd,"Ola KL25Z       ");//completar 16 col com espaços vazios
              LCD1_WriteString((char*)strlcd);
              LCD1_GotoXY(2,1);
         //strcpy(strlcd,"abcdefghijklmnop");
              sprintf(strlcd,"%d",tempo);//completar 16 col com espaços vazios
              LCD1_WriteString((char*)strlcd);
              WAIT1_Waitms(1000);
              tempo++;
       }//FIM DO LAÇO
```



Referências:

Freescale FRDM-KL25Z Referências: www.freescale.com/FRDM-KL25Z.

Post do site MCU on Eclipse de Erich Styger:
https://mcuoneclipse.com/2012/12/22/hd44780-2x16-character-display-for-kinetis-and-freedom-board/

Componente LCDHTA autoria de Erich Styger :
http://steinerberg.com/EmbeddedComponents/LCDHTA/home.htm

Componente WAIT autoria de Erich Styger:
http://steinerberg.com/EmbeddedComponents/Wait/home.htm