UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO DA UFSC
CAMPUS UNIVERSITÁRIO REITOR JOÃO DAVID FERREIRA LIMA - TRINDADE

IMPLEMENTAÇÃO DE UM LETREIRO NA PLACA DE DESENVOLVIEMENTO ATLYS

23/06/15	CARLOS DE SOUZA MORAES NETO	PROF. EDUARDO AUGUSTO BEZERRA
DATA	REALIZAÇÃO	PROFESSOR

Sumário

1 Resumo

O presente documento descreve o desenvolvimento e teste de um letreiro utilizando a placa ATLYS (que simula o processador SPARC LEON) com uma tela OLED. Todo o software foi desenvolvido em c++, utilizando o conceito de fila.

2 Montagem da plataforma

Para a plataforma foram utilizados os seguintes componentes:

2.1 SOFTWARE

- GRMON-EVAL
- ECLIPSE KEPLER
- MINGW-4.4.2- G++
- BITSTREAM PARA A PLACA ATLYS DO LEON3 (LEON3MP.BIT)
- Driver USB-UART da DIGILENT (ADEPT 2).

2.2 HARDWARE

- PLACA ATLYS COM O XILINX SPARTAN-6
- PLACA PMOD OLED PB-200-222.

3 Arquitetura do código

Para organizar o código, foi dividido nas seguintes partes o código:

- NODO.H e NODO.CPP: possui o código do elemento da fila, com todas as funções pertinentes ao elemento;
- FILA.H e FILA.CPP: possui o código para a manipulação de uma fila com as seguintes funções para o elemento fila:

Void inserir(std::string prop)	Inseri novo elemento na fila com o texto prop.
Void retirar()	Retira o elemento atual da fila.
Void próximo()	Próximo elemento da fila. Volta ao primeiro após chegar no ultimo elemento.
Void juntar(fila* nova)	Inseri uma fila inteira a fila existente.
Void limpa()	Apaga o elemento da fila.

- CONTROLE_PROPAGANDA.H E CONTROLE_PROPAGANDA.CPP : possui o código para controle do sistema do letreiro, interligando a operação com a nova fila. Possui as seguintes funções para o objeto controle_propaganda:

Void exibir(fila* prop, int t)	Exibe na tela OLED a fila existente, unindo com a nova fila prop, com cada propaganda sendo apresentada no tempo t).

- Display.h : classe abstrata com as funções que controle_propaganda necessita para sua operação:

Virtual void mostra(string txt)	Exibe o texto.
Virtual void apagar_tela()	Apaga o texto.
Virtual bool apaga()	Retorna o status do botão apagar.
Virtual bool atualiza()	Retorna o status do botão atualiza.
Virtual bool liga()	Liga a tela.

- OLED.h e OLDED.cpp: código fornecido para operar o OLED;
- OLED_driver: classe desenvolvida que herda as funções de Display.h com o objetivo de realizar a interface entre o software do OLED e o CONTROLE do letreiro.
- Teste_tela.h: classe desenvolvida para testar a tela sem o OLED.
- LETREIRO.cpp: código principal com a função main e

4 Fluxo do desenvolvimento

4.1 Uso da std::string

No início todo o software foi feito com o ponteiro char*. Entretanto, quando a informação é aquisitada com cin, não há memória alocada, o que causou problemas com a lista. Para evitar alocação manual de memória com malloc, foi preferível utilizar o std::string.

4.2 Uso do cin;

No início e nos testes foi utilizado o comando getline. Entretanto quando compilado no LEON ocorreu erros de execução.

4.3 Elementos float double;

Por algum motivo o LEON3 não operava variáveis do tipo float e double.

4.4 Time.h

Esta STL operou corretamente na máquina de teste (Windows). No leon o relógio não opera e não possui todas as funções que a STL possui.