Структура данных

Описание активного элемента

typedef struct {

uint16\_t X; // координата по горизонтали

uint16\_t Y; // координата по вертикали

uint8\_t width; // длинна

uint8\_t height; // высота

uint16\_t BkColor; // цвет фона

uint16\_t FntColor; // цвет текста

uint8\_t CanFocus; // может удерживать фокус

uint8\_t txt\_len; // длинна надписи

char \* txt; // надпись

}tActiveElement;

tActiveElement \* Elements; // указатель на массив активных элементов

byte ElementsCount = 0; // количество активных элементов

Функции

void InitDisplay()

Первичная инициализация железки. Поднимается LCD, тачскрин, и карта памяти.

void LoadScreenFromFile(char \* aFileName, uint16\_t aLeft, uint16\_t aTop)

Читаем файл экрана. При этом создается массив активных элементов и на экран выводятся все пассивные и активные элементы.

void LoadScreenSetup(uint16\_t aLeft, uint16\_t aTop) – чтение из файла настроек экрана и прорисовка окна

void LoadElements() – чтение активных элементов

void LoadElement() – чтение одного элемента. Если он пассивный – выводим на экран, иначе сохраняем в массиве.

void UpdateActiveElements ()

перерисовка всех активных элементов экрана в координаты.

void UpdateElement(byte indx)

перерисовка одного активного элемента на экране.

int TouchPress()

обработчик нажатия на активный элемент. Возвращает индекс элемента в массиве или -1, если ничего не нажато.

void DrawButtonBorder(uint8\_t buttonID)

перерисовка рамки активного элемента. Здесь же выделение цветом для «утопленной» кнопки.

void CloseScreen()

закрывает экран. Очищает память от уже ненужных данных.