Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

**МЕХАНІКО-ТЕХНОЛОГІЧНИЙ ФАКУЛЬТЕТ**

Кафедра програмування та захисту інформації

**Звіт**

з виконаної лабораторної роботи № 3

дисципліни “ Інженерія програмного забезпечення ”

на тему:

«Написання специфікації на програму, що розробляється»

Виконав :

студент академічної групи КІ-15

Аннаєв А. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Перевірив :

Викладач

Конопліцька О.К.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кропивницкий 2017

1. **Общее описание**

«Адаптер для тестирования отладочных плат» (дальше как Адаптер) состоит из трех основных комплектующих:

* 1. Программа верхнего уровня, для визуализации и управления процессом тестирования, дальше «Главная форма»)
  2. Pawn cкрипт (скрипт, написанный на языке Pawn дальше «скрипт») для удобства управления ресурсами отладочной платы.
  3. Программа-драйвер для выполнения низкоуровневых операций таких как запуск виртуальной машины и работа с памятью (дальше «Адаптер»).

Использование скрипта облегчает работу при тестировании различного типа отладочных плат.

Визуализация процесса ускоряет выявление дефекта или некорректной работы устройства.

Процесс использования данного программного продукта состоит из возможности загрузки скрипта, как в ручном режиме, так и при помощи отладчика, по последовательному асинхронному порту (дальше ComPort). Загружается скрипт в главную форму, где при помощи кнопок управления производятся такие операции как:

* запуск скрипта,
* загрузка в адаптер,
* аварийная остановка скрипта,
* сохранение результатов теста или получение сведений о некорректном завершении каких-либо действий.

При нажатии кнопки «Запустить скрипт» на адаптере запускается виртуальная машина, которая при помощи скрипта управляет аппаратными ресурсами адаптера и при помощи comport может отправлять и получать данные от тестируемой отладочной платы.

1. **Описание интерфейса**

**2.1. Входные данные**

Процесс запуска состоит из следующих шагов:

* Запуск программы верхнего уровня (Emulator.exe).
* Подключение адаптера к ПК при помощи USB кабеля.
* Запуск Pawn отладчика (pawndbg.exe).
* Загрузка скрипта (test.amx) в ПЗУ отладочной платы.
* Запуск виртуальной машины при помощи нажатия кнопки «Запустить скрипт».
* Подключение тестируемой отладочной платы при помощи коннекторов.

**2.2 Выходные данные**

Результат выполнения теста будет отображен в «Главной форме» в виде текстового оповещения, в котором сообщается о результате успешного или провального выполнения, дополненного техническим отчетом.

**2.3. Описание файлов, входящих в пакет «Адаптера для тестирования отладочных плат»**

В состав пакета тестирования отладочных плат входит:

* Программа верхнего уровня (Emulator.exe).
* Программа-драйвер для отладочной платы (adapter.bin).
* Набор скриптов под различные виды тестирования отладочных плат на языке Pawn (test.amx).

1. **Описание архитектуры**

Как указанно в пункте 1 данной спецификации, комплект для тестирования состоит из трех составляющих.

Программа верхнего уровня

Скриптовой отладчик

Адаптер

Тестируемый блок

Виртуальная машина

Функционал адаптера

**3.1 Программа верхнего уровня (Emulator.exe)**

Программа адаптирована для платформы .Net Framework 4.0 и написана языке С#.

Состоит из следующих модулей:

* Protocol.dll.
* Ethernet.dll.
* Uart.dll.
* ProcessData.dll.
* Main class.
  + 1. **Детальное описание модулей**
       1. **Protocol.dll**

Библиотека, позволяющая формировать пакеты и расформировывать их.

Имеет следующий функционал:

1. Формирование пакет.
2. Расформировка пакета.
3. Преобразование массива байт в строковое представление.

/// <summary>

/// Формирование пакета с подсчетом контрольной суммы

/// </summary>

public void FormingPack(){}

/// <summary>

/// Распаковка пришедшего пакет

/// </summary>

public void UnFormingPack(){}

/// <summary>

/// Преобразование массива байт в строковое представление

/// </summary>

public void ByteArrayToString(byte[] message){}

* + - 1. **Ethernet.dll**

Библиотека, позволяющая отправлять и принимать пакеты данных через Ethernet по протоколу TCP. Имеет следующий функционал:

1. Подключение.
2. Отключение.
3. Отправка данных.
4. Прием данных.

/// <summary>

/// Подключение к конечной точке по протоколу TCp

/// </summary>

public void Connect(){}

/// <summary>

/// Отключение

/// </summary>

public void Disconnect(){}

/// <summary>

/// Прием данных

/// </summary>

public void ReceiveData(){}

/// <summary>

/// Передача данных

/// </summary>

public void TransmitData(byte[] message){}

* + - 1. **Uart.dll**

Библиотека для работы с UART.

Имеет следующий функционал:

1. Подключение.
2. Отключение.
3. Отправка данных.
4. Прием данных.

/// <summary>

/// Подключение к конечной точке по протоколу TCp

/// </summary>

public void Connect(){}

/// <summary>

/// Отключение

/// </summary>

public void Disconnect(){}

/// <summary>

/// Прием данных

/// </summary>

public void ReceiveData(){}

/// <summary>

/// Передача данных

/// </summary>

public void TransmitData(byte[] message){}

* + - 1. **Main class**

Точка входа в программу верхнего уровня.

Имеет следующий функционал:

1. Обработчик нажатия кнопки «Загрузка скрипта».
2. Обработчик нажатия кнопки «Запуск скрипта».
3. Обработчик нажатия кнопки «Остановка скрипта».
4. Обработчик нажатия кнопки «Сохранение отчета».
5. Обработчик пришедших пакетов.
6. Отображение процесса выполнения теста.

/// <summary>

/// Загрузка скрипта в отладочную плату

/// </summary>

public void LoadScriptToAdapter(Object sender, EventArgs e){}

/// <summary>

/// Запуск скрипт

/// </summary>

public void RunScript(Object sender, EventArgs e){}

/// <summary>

/// Аварийное завершение скрипт

/// </summary>

public void AbortScript(Object sender, EventArgs e){}

/// <summary>

/// Сохранение отчета

/// </summary>

public void SaveReport(Object sender, EventArgs e){}

/// <summary>

/// Процесс обработки данных

/// </summary>

public void DataProcess(){}

/// <summary>

/// Отображение процесса выполнения скрипта(теста)

/// </summary>

public void DisplayReport(){}

* 1. **Программа драйвер для отладочной платы (adapter.bin)**

Программа адаптирована для отладочной платы TivaWare tm4c1294ncpdt и написана языке Си.

Состоит из следующих частей:

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Загрузка скрипта

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void LoadScript(uint8\_t \* data , uint32\_t size);

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Запуск скрипта

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void RunScript();

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Остановка скрипта

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void AbortScript();

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Нативная функция аппаратной инициализации

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

static void HardInitialization();

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Конфигурация ethernetPHY

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void EthernetConfiguration();

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Обработчик прерывания приема данных по Ethernet

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void EthernetReceive();

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Отправка данных по Ethernet

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void EthernetTransmit();

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Обработчик прерывания приема данных по UART

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void UartReceive();

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Отправка данных по UART

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void UartTransmit();

* 1. **Pawn script для тестирования отладочной платы (test.amx)**

В данной программе скрипт работает на удаленном хосте с возможностью удаленной отладки.

Состоит из следующих частей:

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Нативная функция аппаратной инициализации

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

native void HardInitialization();

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Нативная функция конфигурация ethernetPHY

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

native void EthernetConfiguration();

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Нативная функция отправка данных по Ethernet

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

native void EthernetTransmit();

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Нативная функция отправка данных по UART

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

native void UartTransmit();

1. **Функциональные требования**
   1. **Требования к программе**
      1. Комплект для тестирования (дальше «Программа») должен иметь возможность загрузки скриптов, написанных на языке Pawn и имеющих расширение \*.amx, в ПЗУ отладочной платы.
      2. Программа при загрузки некорректного скрипта:
   * Должна игнорировать его.
   * Оповестить пользователя об ошибке.
     1. Программа должна иметь возможность запуска виртуальной машины при помощи нажатия кнопки «Запуск скрипта» и после запуска оповестить пользователя об этом.
     2. Программа должна отображать результат каждого этапа тестирования:

«Запуск теста» >> «Аппаратная инициализация» >> «Тест 1», «Тест 2» … «Тест N» >> «Тест завершен успешно/ некорректно» и отображение технических деталей таких как: время выполнения и продолжительность этапа.

* + 1. Программа должна иметь возможность аварийного завершения выполнения скрипта.
    2. Программа должна иметь возможность сохранения результатов тестирования в виде отчета в .pdf или .xml формате.
  1. **Положительные входные данные**
     1. При нажатии кнопки «Подключиться по Ethernet» в нижней части программного окна должно отобразиться «Ethernet: Подключен».
     2. При нажатии кнопки «Подключиться по UART» в нижней части программного окна должно отобразиться «UART: Подключен».
     3. При нажатии кнопки «Загрузка скрипта» в нижней части программного окна должно отобразиться последовательное изменение состояния от «Скрипт: Загружается» (при загрузке) до «Скрипт: Загружен» (при завершении).
     4. При нажатии кнопки «Запуск скрипта» в нижней части должно отобразиться «Скрипт: Запущен», а в среднем окне отображается процесс выполнения скрипта.
     5. При нажатии кнопки «Сохранить отчет» должно появиться диалоговое окно с выбором расширения для сохранения .pdf или .xml.
     6. При нажатии «Остановить скрипт», после небольшой задержки, в нижней части окна программы должно отобразиться «Скрипт: Остановлен», а в среднем окне появляется надпись, оповещающая о аварийном завершении скрипта.
     7. При полном завершении теста в среднем окне отображается «Тест завершен» и технические детали о процессе тестирования. В нижней части экрана - «Скрипт: Завершен».
  2. **Отрицательные входные данные**
     1. При попытке загрузить скрипт, не подключаясь к Ethernet, всплывет сообщение с текстом, предупреждающим о том, что нет подключения Ethernet.
     2. При попытке запустить скрипт, прежде не загрузив его, всплывет сообщение с текстом, предупреждающим о том, что скрипт не был загружен.
     3. При попытке остановить скрипт, прежде не запустив его, всплывет сообщение с текстом, предупреждающим о том, что скрипт не был запущен.