STUDENT NAME _____

MID-TERM REPORT EVALUATION

(1) Technical component:

Advisor _____
Name of Advisor

Executive Summary 3.
5 points maximum; should succinctly state objective and technical approach of project suitable for a person outside the class.

Introduction 4
5 points maximum; should present the basic problem to be solved and relevant background.

Methodology 9
10 points maximum; should describe in detail how the design task is to be accomplished. This should be the longest section and should include references as appropriate.

combined

Progress 5
5 points maximum; should describe the current status of the project using the milestones and timetable in the proposal.

Conclusions 3
5 points maximum; should summarize the report with particular emphasis on what remains to be completed.

Modifications 10.
10 points maximum; should describe any technical changes in the project or changes in the timetable/scope of work since the proposal was submitted.

(2) Communications component

Organization _____
Letter grade only; was the order appropriate, i.e., executive summary first, etc.

Comprehension _____
Letter grade only; can the reader understand what is proposed?

Grammar _____
Letter grade only

Modeling Bioterrorism

A Report of Mid-Term Progress

Kenneth J. Meier

July 8, 2003

Executive Summary

The objective of the project is to ~~achieve~~ demonstrate a software ~~implementation of an existing mathematical model~~ simulation of a theoretical smallpox attack on the city of New York. The modeling software MATLAB ~~was determined to be the superior platform over the alternate Globesight software~~ is used for the simulation. The project is currently in a state of research as only simple MATLAB statements and equations are being tested at this time. After overcoming the MATLAB software's learning curve, implementation of the smallpox model can proceed.

Say something about inputs/outputs, graphics, etc.

Introduction

The purpose of this project is to achieve a software implementation of an existing mathematical model of a theoretical smallpox attack on the city of New York. The mathematical equations comprising the model were created during a previous semester by system and control engineering students.(1) Initially the platform of the project was to be chosen from the software Globesight or MATLAB. It was decided that the project would be implemented using MATLAB. When completed, the model should offer an insightful illustration of the possible implications of a terrorist attack using smallpox.

See Executive summary.

You should say something about the model, i.e., difference equations, etc.

Methodology and Progress

A previous attempt to implement the smallpox model using Globesight platform met with numerous problems, the most fundamental being bugs and lack of support for the software. Upon investigation it was determined that MATLAB is a more suitable platform for the smallpox model, reaching the project's first milestone. MATLAB does not suffer from the myriad bugs plaguing Globesight, and unlike Globesight, MATLAB is a commercially available software package currently in widespread use by engineers. The algorithm to be used for writing the project's MATLAB code is to be adapted from the previously attempted Globesight implementation. Present work on the project is being focused on developing the engineer's proficiency in MATLAB programming. Once this initial learning curve has been overcome, actual coding of the smallpox model will begin.

OK.

Any thought on input/output?

(1) Equations should be in an Appendix
Also dictionary of variables.

Conclusions and Modifications

say. how was this determined?

It has been determined that MATLAB is the superior choice for implementation of the smallpox model, completing the first project milestone. However, the rest of the project's planned milestones have proven overly ambitious. The second milestone is currently in progress, with the algorithm being adapted from the previously attempted Globesight implementation. Actual coding in MATLAB has been limited to simple statements and equations to acquaint the engineer with MATLAB based programming. The remaining project milestones have adjusted by about ten days to the schedule schedule:

I think it would be better to combine progress & modifications.

July 11, 2003 - Algorithm Completed and Implementation Started

July 23, 2003 - Implementation Completed

July 28, 2003 - Debugging Completed

STUDENT NAME _____

MID-TERM REPORT EVALUATION

(1) Technical component:

Advisor _____
Name of Advisor

Executive Summary _____
5 points maximum; should succinctly state objective and technical approach of project suitable for a person outside the class.

Introduction _____
5 points maximum; should present the basic problem to be solved and relevant background.

Methodology _____
10 points maximum; should describe in detail how the design task is to be accomplished. This should be the longest section and should include references as appropriate.

Progress _____
5 points maximum; should describe the current status of the project using the milestones and timetable in the proposal.

Conclusions _____
5 points maximum; should summarize the report with particular emphasis on what remains to be completed.

Modifications _____
10 points maximum; should describe any technical changes in the project or changes in the timetable/scope of work since the proposal was submitted.

(2) Communications component

Organization _____
Letter grade only; was the order appropriate, i.e., executive summary first, etc.

Comprehension _____
Letter grade only; can the reader understand what is proposed?

Grammar _____
Letter grade only

Proposal

Title: 4D/Web-Based Data Acquisition System

Project Engineer: Stephen Watt

Date: 7/7/03

Sponsor: Mr. Mark Podany, PGM Diversified Industries

Executive Summary:

This project involves using a commercial data acquisition system with built in http support as an information server for a 4D application. Currently the DAS system is operational without direct web support. The concept of using 4D as an http client has been found to be possible but no application has been written.

Introduction:

The original problem to be solved was to be able to have remote data acquisition systems (DAS) that could be monitored from a central application written for 4D. The purpose of this project is to be able to integrate a 4D application with a commercially available data acquisition system. The DAS that is being used in this project is a JKMicro Flashlite 386EX board with Ethernet and A/D add-on cards. This board has 8 A/D channels which can be polled by the 386EX board. The board is basically a small computer with its own disk operating system. There are many web based DAS solutions available but what is special about this project is the creation of a custom application in 4D for connecting to the DAS using the http protocol. 4D is a database application platform much like Microsoft Access. It allows for the creation of relational databases and custom queries or forms. For the project a custom form will be created that can connect to the DAS via http, parse the needed information from the served web page, and then display that information in the 4D form.

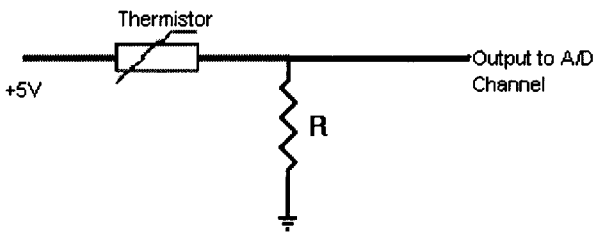
Methodology:

For this project the solution had been defined, so this is more an implementation scheme then a design methodology. To implement this solution I knew there would be a large learning curve as I had never worked with either of the two products before. Therefore the implementation would take on 4 main stages.

- Learning about the DAS
- Writing software for the DAS
- Learning about 4D

- Writing an application for 4D

For the DAS I decided to build a simple temperature circuit to input into the A/D then I needed to write an application that could output that data to a dynamic web page. The basic algorithm for this to work is to get a voltage from the temperature circuit that is relational to temperature. Using that voltage, calculate the temperature, and have this data inserted into a dynamic web page. To get data from the A/D board my application has to first write a control bit to the A/D register on the DAS board. This tells the A/D which channel you are reading and within what voltage range the value should be in. The program then has to wait at least 50 microseconds and then reads the same register for the value on the A/D channel in A/D ticks. Those A/D ticks can then be converted to voltage. Here is the circuit diagram for the temperature sensor.



Once the voltage on the Resistor is know the loop current can be calculated and then used to calculate the thermistor resistance. With the thermistor resistance the program can then calculate the temperature using the characteristics of the particular thermistor. This is a very simplistic circuit buts its purpose is that it gives a somewhat changing signal to the A/D so that later when the 4D application is tested you can see the data change. The next part of solution implementation is to get the data into a web page. The way JKMicro implements getting data into the web server is by way of server side

includes in the html code for the web page. The html would look like the following example.

```
<html><body>
<h1><!-- #include file="0" --></h1>
</body></html>
```

These server side includes tell the web server to get the data from a location in a memory lookup table. The data is put into memory by one of two possible ways. One way the JK micro supports this is to write a custom terminate and stay resident (TSR) application that write the data you want the web page to be able to access to the memory table when interrupt int31h is called. A newer version of the web server has the A/D support built in and can access the data on the A/D port without having a running TSR.

Once that is complete then the 4D application can be written to parse information from that dynamic web page and show the data in a form. 4D has several commands that allow http get calls to web servers. The data can then be parsed and output to a form.

Progress

Currently the JKMicro board is functioning with the A/D. I wrote a small program to use the temperature circuit and then output the temperature to the console (Source Code Attached). When it came time to get the data into the web server several problems came up mostly relating to not having the Borland compiler that JKMicro uses in its example source code. Currently no data is output to the webpage but I have obtained a new version of the web server which will not require writing a TSR and instead can read from the A/D board directly. I should be able to use JavaScript to calculate and output the temperature instead of writing a C application. Nothing has been written for 4D yet but I

have spent many hours familiarizing myself with the application. I will hopefully begin development of the 4D application in the next few days. Much of my time so far has been spent learning how both the software and the hardware operate.

Conclusions:

Currently the project is moving along approximately on schedule. The A/D data needs to be integrated into the web server so a decision has to be made on whether to continue to try and write a custom TSR or to use the built in A/D scripting with the newer web server software. Once that is completed then the Flashlite board can have its final testing. The 4D application needs to be started but I have yet to find a good programming reference for creating custom forms. Much of the progress made has been in learning the two systems and becoming familiar with their operation.

Modification:

With the delays in integrating the A/D output into the web server and the problem finding a good programming reference for 4D then it might be necessary to increase the time spent for both research and programming of the 4D application. Hopefully the decision will be made to use the newer web server which might still allow me to finish on time if a good programming reference can be found. I have included an alternate timeline in case an extension is necessary that which shows alternate finishing dates for possible extended tasks. If a switch is made to the newer web server the application I initially wrote as a compiled C program will have to be redone as a JavaScript application to be run by the

web client or possibly get the data from the web page in a raw format and have 4D perform the necessary calculations.

Appendix: A/D temperature sensor source code in C++:

```
include <stdio.h>
#include <conio.h>
#include <dos.h>
#include <bios.h>
#include <iostream.h>
#include <iomanip.h>
#include <math.h>

#define outportb outp
#define inport  inpw
#define AD1        0x2B0

//#pragma argsused
main (int argc, char **argv) {

int j;
int ControlBase = 0x40;                 // norman operation, internal clock,
                                        //range = 0V - 5V  / A/D port 0
unsigned int volt;
char sign=0;
float voltage;
float current;
float temperature;
int resistor = 2160;
float thermistor;
int thermistor_constant=350;
float thermistor_ratio;
float log_ratio;
float a=3.3540167e-3;
float b=2.6591761e-4;     // Thermistor Constants
float c=1.4616501e-6;
float d=-6.0475280e-8;
float temperature_prev=0;

while ( !kbhit() )
{
            outportb( AD1, ControlBase);    //Write Control Bit to A/D register
                  for( j = 0; j < 100; j++ ) {}  // Count to 100 for a delay

                  volt = inpw( AD1 );          //Read A/D value
                  voltage = volt * (5.0 / 4095.0);
                  current = voltage / resistor;
                  if ( current < .00001)
                  {           cout << "Please Connect Circuit";
                              return 0;
                  }
                  thermistor = (5-voltage)/current;
                  thermistor_ratio = thermistor/thermistor_constant;
                  log_ratio=log(thermistor_ratio);
                  temperature = 1/(a+b*log_ratio+c*(log_ratio*log_ratio)+d*(log_ratio*log_ratio*log_ratio));
                  temperature=temperature-273;
                  temperature=(temperature*1.8)+32;
                  if (fabs(temperature-temperature_prev)>1)
                  {cout << "\r" << setprecision(4)<<temperature << " F"; }
                  temperature_prev=temperature;
}
getch();
return 0;
}
```