

Personal Project Proposal: Intelligent Floor Vacuum

For my individual project, I would like to create a controller for an intelligent floor vacuum. This device will move around a room and clean the floor. By pressing S, M, or L this will tell the vacuum how big the room is to clean and how long to stay on. The floor vacuum has four sensors: a wall following sensor, a cliff avoidance sensor, a virtual wall sensor, and a contact sensor (the bumper). When one of these sensors goes off the floor vacuum will respond accordingly to its input. For example, if the vacuum bumps into an object, creating pressure to its right bumper, it will rotate its wheels 90 degrees to the left and then will proceed forward.

This project will consist of many operations such as waiting to be turned on or off, monitoring the sensors, moving the wheels if one of these sensors goes off, cleaning the room according to the room size, turning off automatically after it has reached a certain time limit (depending on the room size).

I was going to use the PIC board to simulate the random objects, virtual wall, or staircase that the floor vacuum will encounter by pressing certain buttons and having those sensors tell the device what to do.

Excellent, go for it.



Personal Project Proposal: Video Poker

I've never been to Las Vegas so I've decided to bring a little bit of it to SCU. For my personal project I will design and implement the software for a video poker machine similar to those in Las Vegas casinos. A player will deposit money into the system, which will determine how much was deposited and assign a pool of credits from which the player can draw. The player can deposit more money at any time to increase their credit pool, even during gameplay. Before a hand can be dealt, the player must set a bet amount in the range of 1 to 5 credits.

The actual gameplay will function as a sort of single-player 5-card draw poker. When the player pushes the deal button, the system will display five different random cards. The player then pushes buttons corresponding the positions of the cards they want to hold, which become marked. After selecting the cards to keep, the player presses the deal button again to discard the unmarked cards and replace them with new cards. The player's winnings are determined by a preset schedule: three of a kind wins 5x, a straight wins 20x, etc., where x is the number of credits the player bet; the more a player bets, the larger the payoff they receive. The playing cycle can be repeated until the player pushes the cash-out button or runs out of credits.

All of the previous descriptions dealt with the player's side of the system. On the casino side, the system must keep track of the amount of money it has stored internally for players who eventually cash out. On the off chance that the player's credit pool is so large that its cash value is approaching the amount of money inside the machine, the system will automatically cash out to prevent a player getting short-changed. It will signal that it is nearly empty and suspend operation until it is refilled.

I believe that all these functions can be adequately simulated on the systems available. Below are preliminary layouts for the character screen and buttons. Any playing card can be represented by 2 characters (one for the value, the other for the suit), with a third to serve as the hold mark. Even with 5 cards there will still be plenty of room for the current bet amount and the credit pool. Also, there are plenty of buttons for all the functions.

Character Display:

9	H			K	D	*		8	S					B	3
		T	D			K	H	*		C	0	0	1	3	5

Button Layout:

Hold Card 1	Hold Card 2	Hold Card 3
Hold Card 4	Hold Card 5	
Bet One	Bet Max	Deal
Cash Out	Deposit	Refill Cash

Very good, go for it.



Personal Project Proposal

Ceiling Fan

Project: To create an embedded system for a ceiling fan which operates as a heating and cooling system

Components:

- Overheat protection system
- Heater fan
- Automatic Thermostat
- Wall Panel
- Pull chains: Light and Heater

In warm weather, the fan is used like a regular ceiling fan. Since heat rises, there is cooler air at a lower altitude. The fan brings the cooler air from the floor to the ceiling through constant motion of the fan blades. It works independently from the heater. The fan has two heat settings, an adjustable automatic thermostat and an overheat protection system.

In cool weather, the heater is turned on either automatically (by using the adjustable automatic thermostat) or manually (by using the pull chain). For safety purposes, an overheat protection system is installed. It automatically shuts off if there is a malfunction. The heater is placed at the base of the ceiling fan and results in warming the air. The heater fan helps the heat spread throughout the room. There will also be a device on the wall telling the person the temperature of the given room. On this panel, the person can "talk" to the fan. It can tell the fan what to do and when to do it. The person can set a time and the fan will work either as a heater or an air conditioning device.

This looks fine, go for it.



COEN 120 project topic

I'm planning on creating the module for a programmable home thermostat. A home thermostat isn't as simple as it seems. It does multiple functions. The thermostat contains the current temperature of the house when it is off.

A programmable home thermostat basically does 2 functions: heating and cooling the house. On the thermostat there is a sensor that takes the current temperature of the house. (in degrees Fahrenheit) A switch on the left allows the user to control whether the house gets heated (mainly in the winter) or cooled. (primarily in the summer) These functions are located on a left switch in the thermostat in my house. There is a right switch that allows the user to turn the programmable thermostat from off to either automatic or fan. When the user sets the button to heat, (and turns the button on the right to automatic) the default temperature will be shown (normally the setting from which was used earlier) allowing the user to either raise, lower, or to keep this temperature as the desired one. Once the desired temperature is set, the thermostat will compare this desired temperature to the current temperature in the house. If the desired temperature is greater than the current temperature, then the thermostat will heat the house. Otherwise it will not.

The same idea applies to the cooling feature of the thermostat. First the user must set the left switch to the cool function. Then, he/she must set the right switch to the automatic position. A default temperature appears allowing the user to either rise, lower, or keep this temperature as the desired temperature. The thermostat then checks the current temperature in the house and compares it to the desired temperature. If, the current temperature is higher than the desired temperature, the thermostat will turn the air conditioning on cooling the house. Otherwise, it won't turn on.

The fan switch on the right switch isn't dependent on the left switch. The fan function acts as it names implies. It just airs the house out. A fan generally makes a house a little cooler (just a few degrees) while airing it out. The fan takes less energy to run than the heater and air conditioner.

Most thermostats also have a setting so that for certain time intervals, it can be set to a certain temperature. This way the user doesn't have to constantly set the thermostat. The thermostat has a calendar and clock in it. The user is allowed to set the temperature for intervals during the day. I'm going to split the day into four intervals: morning, afternoon, evening, and night. Morning is the time interval from 6 a.m. to 12 p.m. Afternoon is the interval from 12 p.m. to 6 p.m. Evening is the interval from 6 p.m. to 12 a.m. and night is the interval from 12 a.m. to 6 a.m. The temperature is set to 75 degrees in the morning, 65 in the afternoon, 70 in the evening, and 60 at night. The thermostat will check to see whether the temperature needs to be raised or lowered depending on the season.

This is better. You might want to consider letting the user specify the time intervals indicating morning,... also the temperatures that the system used during these times.

COEN 120
PERSONAL PROJECT PROPOSAL:

Answering Machine Controller

For my personal project, I will implement a controller for an answering machine. My answering machine will support a maximum of 10 messages. The messages will be able to be played, erased, and repeated. A user can also skip (or fast-forward) a message to listen to the next message on the machine. A greeting will also be given when the answering machine picks up, telling the person on the other end to leave a message.

The answering machine controller will contain two threads: the answering thread, and the message playback thread. The answering thread will loop continuously, waiting until the phone rings. Then it will “pick up,” a greeting will then be given, and then the message will be recorded. When the person on the other end of the line hangs up, the answering thread will again wait.

The playback thread will be responsible for performing various functions on the messages. If “Play” is invoked, all the messages in the answering machine will be played, starting with the oldest message (if no messages are on the machine, nothing will happen). If “Stop” is invoked, the messages will stop playing (if they are not currently playing, nothing will happen). “Repeat” will play the previous message (if the last message is being played, or no messages are currently playing, nothing will happen). “Skip” will stop playing the current message, and play the next message (if the last message is being played, or no messages are currently playing, nothing will happen). If “Erase” is invoked while no messages are playing, all the messages on the machine will be erased (if no messages are on the machine, nothing will happen). If a message *is* playing, “Erase” will erase the message that is currently playing.

Given time, I would like to implement the following features:

- Allow recording of a personal greeting
- Allow a user to specify the number of rings the answering machine should wait before picking up
- Support for an unlimited (or at least larger) number of messages

Good project, I suspect that you will simulate the message with text. Go for it.



Coke Machine (Bottled Drink Dispenser)

(Note, all ideas marked with * are for a possible addition if it needs to be more complex)

Input

A number of binary input buttons equal to the number of unique beverages held, plus a cancel/change return button.

A coin slot that can distinguish what coins have been used and send any invalid ones to the change slot.

A reader that will accept only actual dollar bills (which increments it will recognize is subject to change).

*[An ACCESS card reader that can get a student ID number (possibly simulated by entering the card number on a PIC?)]

Sensors

Some type of sensor will be needed to detect whether there is a bottle currently held in the lowest position of each bottle stack.

Thermometer to measure internal temperature.

Actions/Events/Misc

- Whenever money is entered through the coin slot or the dollar bill reader, the amount will be stored as a float or double and the machine's display will show the current amount of credit. *[For the ACCESS card, the machine will contact a simple database of account numbers and account balance information. It will check to see if it is a valid card and if the account has enough money to be deducted. If so, a pre-set increment (probably a dollar), is credited.]
- Whenever a beverage button is pressed, it will check the sensor for that stack to make sure there is a beverage that can be dispensed. If so, it then checks to make sure there is sufficient credit. If both of these are true, the beverage is released and the machine cycles the bottles in that stack. The cost is subtracted from the credit and the correct change is returned. *[Any credit remaining from the ACCESS account is re-added to the balance of that account.]
- If the cancel button is pushed at any time, it will exit any routine (except maintenance and cooling) and return the credit by the methods outlined above. After, it will resume waiting for input.
- The cooling routine would be a constantly running thread, which regularly checks the temperature. It would probably only need to do so once a minute, as the temperature is expected to change rapidly, nor is the cooling vital (though most people wouldn't buy a warm Coke). The temperature would have an upper bound of, say, 70 degrees Fahrenheit. If the temperature exceeds this amount, the machine will trigger a cooling mechanism. It will continue to cool until it reaches a lower limit of 65 degrees and then shutdown the cooling device. The regular temp checks would occur throughout the cooling process. One possible extra would be to check the stack sensors before initiating the cooling process and not to execute it if there are no beverages.

- The maintenance routine would be caused once a service attendant opens the machine. The cooling routine would probably be left active during this, but all transaction processes would pause.
- *[The ACCESS card database would be run on a separate target machine. The dispenser could communicate with it via the following functions: credit, debit, getBalance, and isValid.]

This is an excellent project. You can forget the ACCESS card, you will have enough to do without it.



COEN120 Project Proposal

“Treadmill Controller”

For individual project I am planning to create a system that would be able to control a treadmill running machine. Every decent **jym** has few treadmills that allow people to run indoors. Treadmills have an embedded system running within it providing the following services:

- Turning on and off the treadmill
- Setting the desired speed
- Setting the desired incline of running
- Setting the length of running session
- Various pre-made running practices
- Information about current system settings
- Slow “cool down” mode

In order to accomplish users demands embedded system uses inputs from the following sensors:

- Several buttons
- Sensor for weight
- Sensor for heart rate

Treadmill's control system also needs few security measures like steady increment of incline, auto switch off if nobody is running on it, immediate switch to "cool-down" mode etc.

During a typical use of the treadmill, user needs to turn it on and choose between manual and predefined running settings. After that thread gradually adjust speed and incline until it matches the users settings. If the user grabs the treadmill's handles the heart beat sensor measures his heart rate, notifies him about his heart rate and maybe even do something more with that data. Shortly before the end of practice the system notifies the user that it will slow down and turn off soon.

Excellent, go for it.



I was thinking of doing something like a pc hardware monitoring system. Things that could be monitored and controlled:

CPU temp
CPU fan speed
System temp
System fan(s) speed(s)
other various temps

It would have lcd screen output that would display the data from the list above. and various knobs that would be used to control fan speeds. The lcd would also display other information such as CPU speed, total RAM, date, and time.

In principle, the project is interesting, but you have WAY under-defined the scope. I want you to define EXPLICITLY what the monitoring device does. You should plan on varying the fan speed depending upon the CPU temperature. (My computer does that!) I do not just want you to display a given set of inputs. Your system MUST react to these inputs and take any appropriate action if necessary. PLEASE RESUBMIT.



NOT APPROVED

Project Proposal

For my project, I plan to create a toaster oven for the public that is easy, safe, and reliable to use. It will supply the needs of customers who want to quickly toast a bagel, keep their lunch warm for a later time, or bake a small dish of lasagna. Thus, the toaster oven would include four different settings on how to cook the food such as: toasting, keeping warm, broiling, and baking or roasting (all self-explanatory). All the customer has to do is turn the knob to the appropriate cooking method and it will do what its specified to do. When the toaster oven is not in used or they want to abruptly stop cooking their food, the customer can simply set the knob to the off setting or they can just unplug the cord. The customer will know if the toaster oven is on because there would be a little red light to signal that it is on. It will stay on depending on the time limit the customer set it out for and when it is finished, the light will turn off. In addition to the red light, there would be an alarm or buzzer of some sort when the timer runs out so that if the customer is not in the proximity of the toaster oven, they can hear that it has finished its task.

In addition to the four different cooking methods, the customer has to control the temperature. This knob would have incremental temperature settings of 50 degrees Fahrenheit till it reaches 450 degrees. Furthermore, next to a certain temperature, there would be appropriate and helpful labels for the various cooking methods such as 250 degrees for keeping the food warm, 350 degrees for baking, and 450 degrees (maximum temp) for toasting quick or broiling. Last but not least, the customer can adjust how long

they want the current cooking method to run for. The time knob would have increments of a minute each. The max setting for how long a cooking method can be is 60 minutes. Also, the customer can cut the time short by just turning the knob to 0 minutes or to the appropriate length of time they feel is sufficient enough. Once the time is up, a buzzer or alarm will go off and alert the customer that the operation has been completed.

What would happen if a fire occurs? Well, if this happens, the alarm or buzzer could and should (hopefully) go off and alert the customer. That or the smell of the burning food would. Another thing that it could do if this occurred is it could turn off (cool off) automatically by lowering the temperature to 0 degrees (I don't know if this is possible yet). All in all, the toaster oven I propose should make a convenient appliance for anyone in need of help in the kitchen or looking for a quick way to cook food.

Looks very good, go for it.



Personal Project

Automated Insulin Injection Device

When designing an automated insulin injector device, we need to take into account several ideas. Insulin helps to regulate the blood sugar levels of a person. In a person without diabetes, the pancreas is the organ of the body which will recreate this. Recreating an organ of the human body is a difficult thing, as there are a lot of “triggers”, which will change the blood sugar of the subjects, such as food, physical exercise, stress, sleeping habits, weight, etc. Unfortunately, we cannot take into account all those triggers, as first of all, they might not trigger the same reaction in blood sugar level in all people, and it is not possible to measure all of those. Today the way to treat diabetes is to inject an amount of insulin which is evaluated with the help of your blood sugar level, as well as the food that you might plan to eat, and the exercise. Furthermore there are further type of insulin which are long acting (action period starting 4-6h after the injection to 10-12h), and short acting (15 minutes to 4h).

In order to design an automated insulin injector device, we need to make the following assumptions:

- a) it is connected to the person 24 hour a day, (inside or outside)
- b) it uses only short acting insulin or Humalog, as it is easier to control, and the use of long acting insulin is to allow one less injection – since the device will be connected to the user all the time, we don't need to worry about that
- c) It will use a sensor that will give a measurement of blood sugar of the user every 15 minutes. Since it is not possible to simulate an entire day to test the device using the pic board, otherwise we would get bored, I have decided to divide the time by a factor x.
- d) It will be able to trigger an alarm.

Let's first talk about the sensor measurement of blood sugar. Blood sugar is normally between 80 mg/dl and 120 mg/dl. If the blood sugar of the user drops below 60 mg/dl it will trigger a sound alarm, to warn the user to eat some sugar. If blood sugar comes in to be more than 180, then it will check when the last injection of insulin was. If the new measurement is between 15' and 2 hours after the last injection, it will do nothing, if it is between 2 hours and 4 hours after the last injection it will then inject half the insulin it is supposed to. After 4 hours, the full amount of insulin is injected. The amount of insulin to be injected is determined by $(\text{blood sugar measurement} - 180) / 15 + 2$.

In case the blood sugar level is measured to be more than 300, then 15 units will be injected once and nothing during the next 4 hours.

The second thing to talk about is an user input which allows to turn the device in “monitor mode” that is not inject any more insulin, but still monitors blood sugar, and gives alarms. (This mode is useful for physical exercise for example).

It uses a clock, to set two modes: day or night; during the night from (midnight to 8am) or user defined the injections will only occur every 4 hours.

Finally an user input device, will allow the user to inject himself an x amount of units of insulin at any time.

I hope this is an adequate project proposal. I am wondering how to simulate the blood sugar levels.

This is excellent, go for it. Just make sure that you are ON TIME with all other aspects of this project.



Personal Project Proposal: Toilet Seat Bidet

For my personal project, I would like to design the embedded system to control a toilet seat bidet. A toilet seat bidet is a modern answer for personal cleanliness. This particular bidet provides a number of features to help clean private areas, post bathroom use.

The Sit On Switch will use sensors to turn the system on automatically as soon as a person sits down on the toilet seat. By this, I mean that the water temperature and seat temperature will begin to increase a default temperature, possibly somewhere around 70 degrees Fahrenheit.

When the user gets off of the toilet seat, the power saving feature will set the water and seat temperature to a pre-heating level. This should be done 10 seconds after the user gets up off of the seat.

The bidet can be set to clean personal areas after using the bathroom with an on/off switch. This particular mode of cleaning should be defined to 2 different modes, one for men, and another for women. Thus, the men's mode will set the tip nozzle closer to seat in order to clean the private regions around the buttocks. Women will need to use this mode as well in order to clean that area. The women's mode of cleaning, however, will extend the nozzle tip further so that the vaginal areas can be cleansed after bathroom use. During use, the water temperature should be able to be set to cold, warm, or hot water. The strength of flow can also be set to low, medium, and high. If the strength of

flow is changed during used, if any predefined cleaning settings were being used, they will automatically go to the general setting for the remaining time.

In regards to the predefined cleanings settings, the user should be able to choose from a variety of cleaning settings for their use. One setting would be for general cleaning. This setting will use the default cleaning time, warm water temperature and medium pressure for cleaning. If the temperature or flow of water is changed, the system will resume using that setting for the time remaining. In addition to this particular setting, the massage setting will allow for the various changes in water flow to simulate massaging. One setting would change from low to medium to high in intervals of 2 seconds. Another from low to medium to low to medium, in 2 seconds intervals, and then to high for 5 seconds. Another one could go from any flow setting to off, and then on again in 2 second intervals.

Cleaning time will be set to a default 1 min. If the user would like to extend the cleaning time, they can input the number of seconds to be added to the clean time. Should the user get off of the seat, the cleaning system will shut off automatically. The nozzle will go back to its hiding position after 10 seconds.

In addition, there will be a warm air drying mechanism with adjustable air temperature and flow. This mechanism will be turned on and off by the user. Should the user get off of the seat, the warm air drying mechanism will be shut off automatically. The air temperature can be set to cool, warm, or hot and the air flow should be able to be set to low, medium, and high.

The position of the nozzle will be able to be changed, during cleaning. Although the system will have default male and female settings for the nozzle, during cleaning, the user should be able to adjust the position of the nozzle.

The toilet seat has adjustable seat warmers to keep your bottom warm. This can be set to off, low, or high.

An example of this product can be seen at www.niceoutlet.com/bidet4.htm . Please let me know if this project is possible to complete, or whether or not it may be too simple or difficult to complete in the allotted time we have. If there are any more questions or concerns, please do not hesitate to contact me.

This is doable and should be fine. You have many little items that the system does in response to events. Think about how you will accomplish this. You may want to consider a table driven method.

This is fine, go for it.

COEN 120
1/19/2003

Personal Project Proposal

For my personal project I want to do a home automation controller, that will control doors, lights, thermostats, and sensors, among other things.

An example of this home automation would be a person comes home at night, presses the button on their garage door remote, the garage door opens, the lights in the garage turn on, the door to the house opens, and the lights in that room in the house turn on. This would be useful if you just came home from the grocery store and you have your hands full with bags of groceries.

Or another example is to have sensors, like motion sensors, that would open the door, and turn on the light, if it was dark. And some rooms could perform different functions upon entry, i.e. if you walk into your home theater room, your television, receiver, etc. would turn on, and the lights would dim, depending on preference.

In each room, there will be a small panel located next to the light switches, for manual use, that would allow you to have settings for that particular room, and there would be a master control panel that will enable or disable the automation of certain rooms. The sensors around the house will trigger a series of actions, programmed by the home owner.

Revision: A set of actions could also be programmed to function at a specified time. For example, it's 8 pm on Sunday night, which is usually movie night for you. So you program your living room to open the door, dim the lights, and turn on the television. Or you have to get up at 6 am every morning, you can schedule a set of commands to turn on your bedroom lights and set off an alarm, or turn on a radio.

Coen 120
Personal Project Proposal

For my project, I would like to design a room controller for a master bedroom. The room controller would be like a remote control for many of the electronics in the room. It would control all of the lights in the room, as well as a fireplace, ceiling fan, sound maker and maybe the curtains. The remote would have buttons for most lights, the fireplace, the curtains and the ceiling fan. The curtains could either be opened or closed. The (ceiling) fan would have three speeds. If the fan was off and its button was pressed once it would go to low speed, another press would make it go to medium speed, a third press would make it go to high speed and a fourth press would turn off the fan. The sound maker and two of the lights would be controlled from analog inputs. The lights would dim and brighten depending on the analog input and the sound maker would get louder or quieter. The controller would have two different functions: manual and program.

The user could manually adjust all of the objects in the room from a user interface. The user could turn on and off the lights by pressing their corresponding buttons. For analog devices like dimmer lights and the sound maker, the user would turn on the object and then use the analog input to control its level. Turning the knob while the object was off would do nothing. But, when the object was turned on it will be operating at the level of the analog input.

The controller would have the capability of programming in different actions to occur at different times of the day. The controller would save 3 or 4 functions. The functions could either be performed daily, or disabled. For example, the user could program the controller to turn on the fireplace and fan to warm the room at 7:00 PM, or set an alarm clock function. The alarm clock function would slowly turn on the analog lights, slowly increase the volume of the sound maker, and finally open the blinds to let the sunshine in.

The room could also contain sensors by the door and fireplace. The door sensor would sense someone entering the room and turn on lights and the fireplace sensor would immediately shut off the gas to the fireplace if an object crossed the sensor.

Excellent Project, go for it.



Professor Quinn
COEN 120
1/23/03

Personal Project Proposal

I plan on making a massage chair similar to the chairs found in The Sharper Image and in Brookstone. This massage chair will massage both the upper body, which would be the neck and the upper and lower back, as well as the lower body, which would be the calves and feet. This massage chair will have four different types of massage: rolling, kneading, percussion, and compression. The massage chair will have different settings that would accommodate the user of the chair. In one of the settings the user will be able to isolate the massage to a specific part of his/her body such as the neck, upper or lower back, calves, or feet. Another setting would be the massage going up and down the body or just massaging different areas of the body at the same time. In these settings the user will be able to control what type of massage he/she wants be it rolling, kneading, percussion, compression, or varied, as well as how fast he/she would like the massage to be. While receiving the massage, the user will be able to stop the massage at any time, pause the massage at any time, or resume the massage after pausing it.

Looks fine, go for it. Simulate the different actions by the use of the lights on the PIC board.



Thursday, January 23, 2003

COEN 120, Professor Quinn

Personal Project Proposal

I plan to create an embedded system that monitors the thrust and maneuvering system for a rocket-propelled missile. The program will receive input, such as desired speed, desired elevation, and desired orientation, and will maneuver accordingly. The system will have to take into account various factors. For example, when controlling the amount of fuel the rocket engine receives, the system will have to monitor the current amount of fuel left. The more fuel within the missile, the more fuel will be required to propel it. The following formula must be applied:

$$a = \dot{m} \cdot V_e / m(t)$$

where 'a' is the acceleration of the missile, ' \dot{m} ' is the propellant flow rate, ' V_e ' is the exit velocity of the propellant, and ' $m(t)$ ' is the total mass at time 't'.

Moreover, the overall drag force of the air in which the missile travels through must be taken into account. This can be included using the equation:

$$F = \dot{m} \cdot V_e + A_e (P_e - P_a)$$

where ' A_e ' is the area at the exit plane of the nozzle, ' P_e ' is the rocket exhaust pressure at the nozzle exit plane, ' P_a ' is the outside atmosphere pressure, and ' F ' is the thrust. Using these equations, and by calculating the necessary force required to propel the missile, we can find out what the necessary propellant flow rate is, which can be controlled by the embedded system. The overall system could be simulated using a standard PIC board. The user could input the requested altitude and orientation using the keypad, and the

system would manipulate the flaps on the missile accordingly by displaying its operations on the computer screen. The user could also input the initial fuel level and the desired speed, and the system would display the current fuel flow rate on the screen as well. Moreover, the user could also input the atmospheric pressure and the nozzle area using the knobs on the PIC board, and the system would compute the drag force exerted and modify the thrust accordingly to maintain the inputted speed.

Great, go for it.





1/23/03
COEN120

Personal Project Proposal rev1: ATM with Targeted Advertising

Since I'm very bad at coming up with interesting ideas, I think I'm going to just try and model a plain, boring ATM. To keep myself interested, I would like to try and integrate an advertising system to complement the normal ATM functions, as seems to be the current trend; the lengths to which companies will go to gain product impressions is frequently amusing, and even though most current banking privacy policies don't permit for it (as far as I can tell), I think that using purchase trends to more effectively target these impressions is an interesting-if rather unoriginal-idea.

As I see it, the system would break down into four areas:

The interface between the customer and the ATM

-This would include the development of a menu for customer input and interaction (a multi-level text menu, I would imagine at this point, navigated by a number pad and menu buttons as is the typical ATM model), as well as the accepting of deposit envelopes and the disbursement of cash. There would also need to be some sort of status menu for ATM techs, to verify system status and to initiate and complete cash replacement and deposit retrieval

The monitoring of the status of the local machine

-This would entail the machine's awareness of how much cash reserves it currently has on hand, as well as the number and age of pending account deposits. Once either state exceeded defined limits, it would notify the central control, which would trigger the dispatch of a tech or whomever to pick up and replenish the respective stocks.

Coordination of account information with a centralized database

-Obviously the account information would have to be stored and retrieved from some remote location, and so the local machine would have to have mechanisms with which to request account status on account activity, as well as a means by which to update the account with the results of recent transactions. As far as modeling this central database, well, I'm not doing this as an exercise in DB design, so I envision this as a pretty simple, low-scale affair, just enough to allow the machine to make a request for account information, and to allow it to update the information serverside. Ideally, this would involve some sort of truly remote arrangement (e.g., a simple TCP setup between client and server PCs), but I'm not sure I'd end up swinging that level of separation in time (I'm open to opinions on how hard that would actually be).

The targeted advertising portion

-As I see it, each account could be classified serverside into (a) different "type(s)" (based on the purchase history a bank would have), and advertising bodies would purchase advertising exposure (in terms of percentage of ads shown for a type, perhaps?) based on these types. When a transaction takes place, the ATM would retrieve the customer type along with the account info, and then schedule a small advertisement (text, in this case) to display during or after the transaction. In addition, there would be arcade-like "attract mode" ads that display in-between transactions.

I'm completely without a sense of perspective-too much? Too little? Vague?

About the right amount. If you store the centralized database on the same machine. Maybe a bit much if you use a network to store this info. The system has storage capability, so you could use the disk for storage. Looks good, go for it. (you may want to display the ads on a random basis)