List of figures

Statement of the problem

The title of the project as presented is “Development of a Tele-control Home Appliance Controller”. It involves the design and development of a remote control unit for controlling different devices connected to it. The remote control unit should use the telephone lines as a medium between the user and the remote devices that need to be controlled. It is a hardware piece of equipment that can be used to switch on and off different apparatuses. As an extension it could report its status to the users and it could be controlled using a personal computer.

Objectives of the project

The main objective of the project is to develop a microprocessor based tele-control unit. The tele-control unit will accept control signals via a telephone line and control the appropriate units connected to the controller. The controller needs to be designed in such a way in order to control commonly used household appliances such as lights, cooker, heater, air-conditioner, etc.

The hardware controller that needs to be produced must be as much professional possible and it must be based on intelligent and professional design. It must meet the requirements and include a research of the subject in depth. It should be as flexible as possible, in order to adapt to the requirements of the supposed client. It is very important to be friendly and handy to the user (e.g. using displays and indicators where possible) so that the user does not have to have any particular knowledge of the system in order to understand its operation. The cost of the device should be kept as small as possible to make it efficient, but this has to be done without compromising the operation and main function of the circuit.

Another goal is also to practise and learn new things related to hardware development by doing a research in depth. At the completion of the project the student will have gain the experience of how is it to design and undertake such a big project.

This project must be as much professional as possible and show a fully understanding of the problem and presenting solutions to this. It can be used as a proof of the student’s work and his knowledge to the subject, later on in his career. It must be a way of showing to the employers his capabilities and what are the things he is able to do in hardware designing.

Relevant background

In this section an overview of the background to the project, including relevant theory and description and evaluation of the resources used is done. The theory and function of some basic blocks used in the project is discussed.

DTMF tones

Telephone service and application vendors discovered DTMF control in late 70s. Vendors standardized the use of DTMF for application control during calls because they were a means of simple user input mechanism and they where standard across all terminal devices.

DTMF means Dual Tone Multi-Frequency. Each DTMF tone is a sum of two sine wave tone frequencies. In DTMF there are 16 distinct tones. Each tone is the sum of two frequencies; one from a low and one from a high frequency group.

The phone only uses 12 of the possible 16 tones. There are only 4 rows and 3 columns of buttons. The rows and columns select frequencies from the low and high frequency group respectively, as shown on figure 1. In a telephone apparatus the “A”, “B”, “C” and “D” buttons do not exist and so these tones are not used. But these are extensively used on electronic keypads.

Frequency (Hz)	1209	1336	1477	1633
697	1	2	3	A
770	4	5	6	B
852	7	8	9	C
941	*	0	#	D

Figure 1. Table showing the keys and frequencies relationship.

Each key is specified by its row and column locations. For example the key "9" is row 2 and column 2 and has a frequency of 852 + 1477 = 2329 Hz. On figure 2 there is a captured screen from an oscilloscope. It is a plot of the tone frequency for the "1" key.

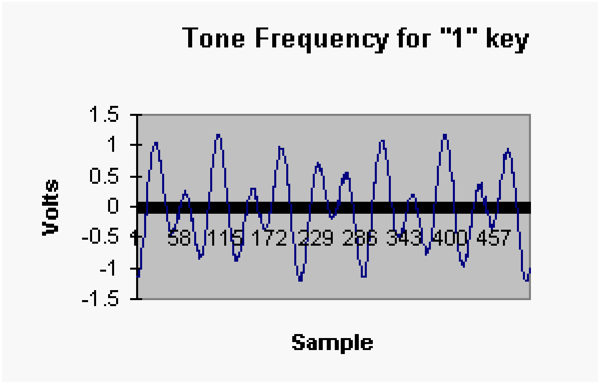


Figure 2. Tone frequency for the “1” key.

It can be seen that the DTMF generated signal is very distinct and clear. The horizontal axis is in samples. The frequency of the tone is about 1906 Hz.

Telephone line wiring

There are differences between telephone wiring used in some countries. The following information shown on figure 3 is web based and shows these differences, and thus the connections that must be done in each country.

Pinout of the RJ11 phone connector
(6 pin with only the inner 4 pins used)

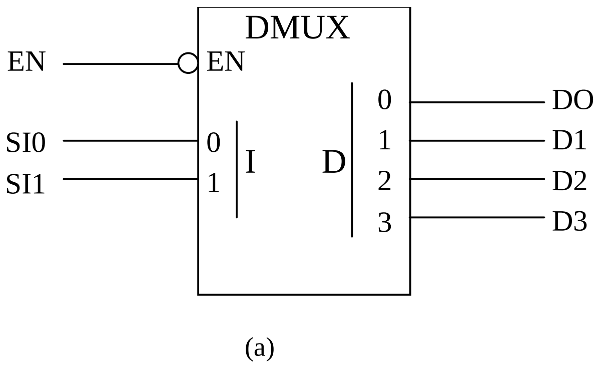
<table>
<tr><td>French (UK) pinout</td><td> a   b
+-||||-+
| |
+-------+</td></tr>
<tr><td>German/Austrian/US pinout</td><td>   ab
+-||||-+
| |
+-------+</td></tr>
</table>

Figure 3. Telephone wiring in different countries.

US/German/Austrian phones just use two wires - the middle 2 pins of the RJ11 connector (= a/b line) and the ringer in each phone is connected across the pair with a capacitor (inside the phone) in series to block the DC path. French (and UK) phones just have one capacitor for the line which is in the wall socket and run a third wire round all the hones.

Demultiplexer

A demultiplexer (DMUX) is a circuit that receives a signal on a single input line and directs that signal to one of 2^n possible output lines. If the enable input is active, we can also call this circuit a n-to2^n decoder.



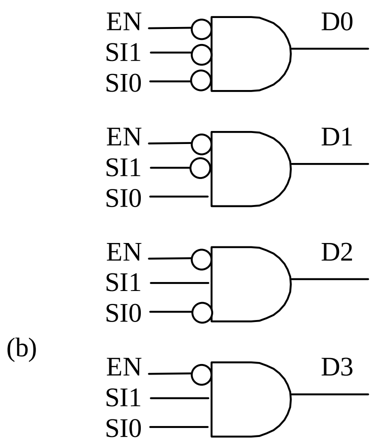


Figure 4. (a)Block diagram for a 1-to-4 line demultiplexer and (b)its functional logic diagram.

EN	**SI1**	**SI0**	**D0**	**D1**	**D2**	**D3**
1	X	X	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	1	0	0
0	1	0	0	0	1	0
0	1	1	0	0	0	1

Figure 5 Truth table of a 1 to 4 line demultiplexer.

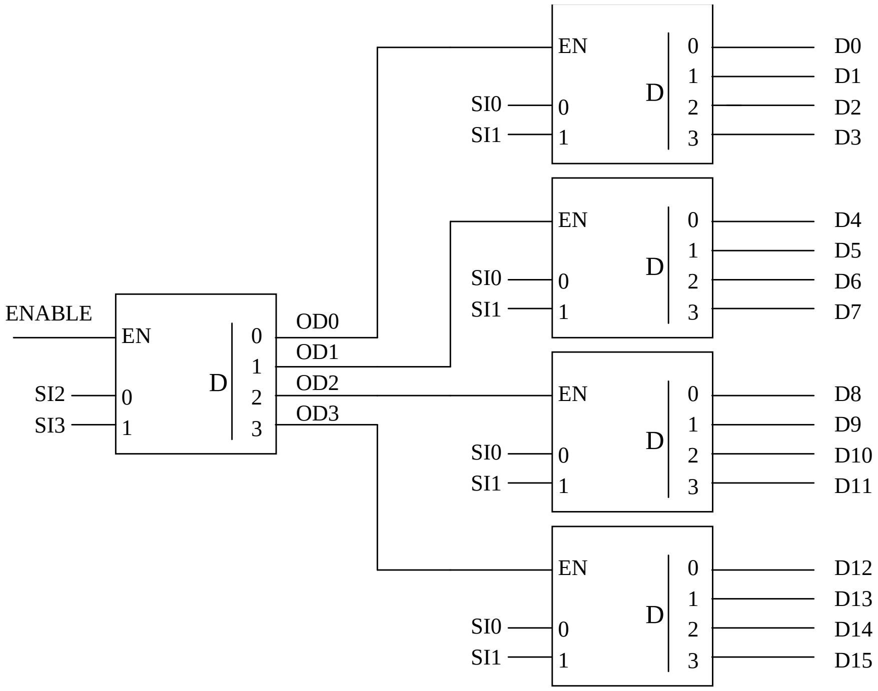


Figure 6. Combination of smaller demultiplexers to form a larger one.

Smaller demultiplexers can also be connected together to obtain larger configuration. The figure 6 above illustrates in block diagram form how five 1-to-4 line demultiplexers can be connected to obtain a 1-to-16 line demultiplexer.

Seven segment display

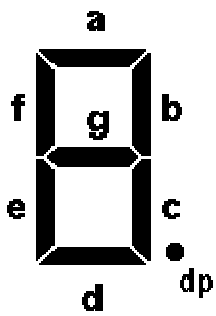


Figure 7. Seven-Segment Display Layout.

The illustration above (figure 7) shows the basic layout of the segments in a seven-segment display. The segments themselves are identified with lower-case letters "a" through "g," with segment "a" at the top and then counting clockwise. Segment "g" is the center bar. Most seven-segment digits also include a decimal point ("dp"), and some also include an extra triangle to turn the decimal point into a comma. This improves readability of large numbers on a calculator, for example. The decimal point is shown here on the right, but some display units put it on the left, or have a decimal point on each side. In addition, most displays are actually slanted a bit, making them look as if they were in italics. This arrangement allows the engineer to turn one digit upside down and place it next to another, so that the two decimal points look like a colon between the two digits. The technique is commonly used in LED clock displays.

Figure 8. Seven Segment Display picture.

Seven-segment displays can be constructed using any of a number of different technologies. The three most common methods are fluorescent displays (used in many line-powered devices such as microwave ovens and some clocks and clock radios), liquid crystal displays (used in many battery-powered devices such as watches and many digital instruments), and LEDs (used in either line-powered or battery-powered devices). However, fluorescent displays require a fairly high driving voltage to operate, and liquid crystal displays require special treatment.

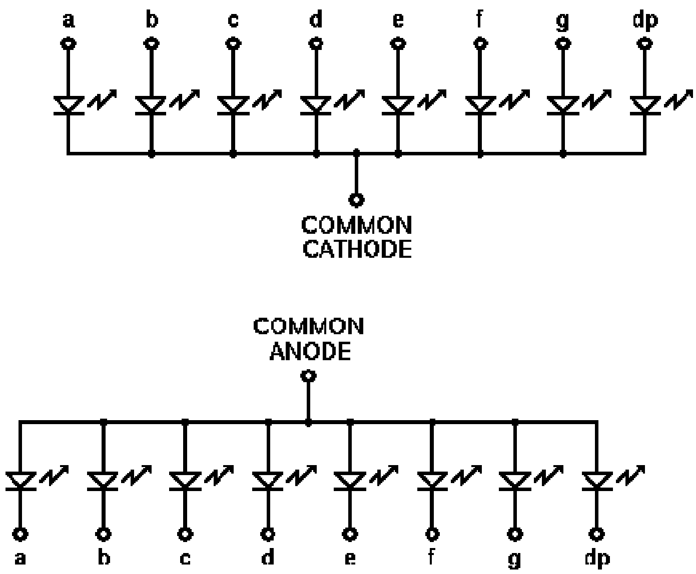


Figure 9. Common cathode and common anode seven segment display schematic diagrams.

As shown in the two schematic diagrams on figure 9, there are two types of seven segment displays; common anode and common cathode ones. The LEDs in a seven-segment display are not isolated from each other. Rather, either all of the cathodes, or all of the anodes, are connected together into a common lead, while the other end of each LED is individually available. This means fewer electrical connections to the package, and also allows us to easily enable or disable a particular digit by controlling the common lead. (In some cases, the common connections are made to groups of LEDs, and the external wiring must make the final connections between them. In other cases, the common connection is made available at more than one location for convenience in laying out printed circuit boards. When laying out circuits using such devices, someone simply needs to take the specific connection details into account). There is no automatic advantage of the common-cathode seven-segment unit over the common-anode version, or vice-versa. Each type lends itself to certain applications, configurations, and logic families.

Optocoupler

Optocouplers are devices that are used for isolation purposes. They can protect sensitive circuits from high currents and voltages. An optocoupler usually consists of a light emitting diode and a photo-transistor. During the operation, the LED emits light (or infrared radiation) and the light hits the base of the photo-transistor. Then the transistor behaves like a closed switch and conducts current. The important bit is that the transistor is not triggered by an incoming voltage (bit) but by the light; thus the output of the circuit is electrically isolated from the input. On figure 10 an optocoupler schematic diagram is shown.

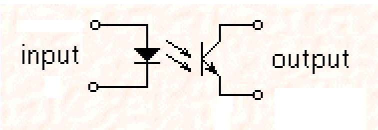


Figure 10. Optocoupler schematic diagram.

Case scenarios

The controller is very useful in lots of applications. It can be used to control household appliances such as lights, cookers, heaters, fridges, TVs, air-conditioners, etc. It can also have commercial use. For example it can be used to control computer servers and other facilities, or to control alarm systems through the telephone line. In industry it could be used to control factory or laboratory switches, motors and other mechanical devices.

For example someone can imagine the case of a person who leaves home to go to work. Accidentally, when he was making his breakfast, he forgot his cooker on. He can’t leave work just to switch off the cooker. By using this controller he can call home from his office, press the relevant button and switch the cooker off.

Another more critical case could be the control of a burglar alarm system. Imagine a family that leaves home for summer vacations and has forgotten to activate the burglar alarm system of the house. They can call from their mobile phone, press the relevant button and switch the burglar alarm on.

Combined with a computer system, the controller could be able to control mechanical or electrical devices in a lab or in a factory. Imagine the use of the device in robotics. By connecting the controller to a computer someone can write a program which would activate and deactivate the appropriate switches at different times and so, it could control the movements of a robot precisely.

The controller could also be used to control radio repeaters or weather stations in remote locations, computer servers etc. Consider the case that someone who is in vacation in Greece wants to switch on a computer server located in Japan, which has been switched off after a power failure. The cost of such a failure could be very high, especially if the server holds user accounts or web space. By having the controller connected to the server he could reactivate the computer in no time. In fact the uses of the device are unlimited and someone can think of so many cases that it would be useful.

Tackling the problem

Description, analysis, discussion and results achieved

Several things have to be considered before starting the project. First of all the signals that will be used to control the device connected to the telephone line have to be determined. Almost all of the modern telephones today support DTMF tones. DTMF tones are the tones used in telephones for tone dialling. Each DTMF tone is a sum of two sine wave tone frequencies. DTMF means Dual Tone Multi-Frequency and there is no baseband multiplexing done on DTMF signals. As mentioned before, the signal generated by a DTMF encoder is a direct algebraic summation, in real time, of the amplitudes of two sine (cosine) waves of different frequencies. i.e. pressing '1' will send a tone made by adding 1209 Hz and 697 Hz to the other end of the line.

There are two basic options of decoding DTMF tones. The first approach uses a DTMF decoder chip that receives audio DTMF tones and decodes them into a sequence of bits at its outputs. There are ready-made very cheap DTMF decoder chips in the market and the right one has to be chosen. This is probably the most critical part of the project, as it is the core that converts audio signals into data bits for further processing.

Alternatively DTMF decoding can be done in software. This requires the use of a microcontroller and the appropriate software to be written to decode DTMF tones. Although this could add more features to the device it is a hard an expensive way to go. As engineers we know that the cost factor is one of the most important ones each time we design a device. Another way to decode/produce DTMF signals is by applying a DSP, a Digital Signal Processor, but DSPs are also quite expensive devices.

Another thing that has to be considered when using the software (microcontroller) solution is that a really good program has to be written and extensively tested in order to ensure that it is bug free and it operates well in all circumstances. My previous experience in programming revealed that there is no absolutely bug free and secure software, or it is difficult to create one. By using software someone can more easily do things compared to hardware but pure hardware can be more deterministic.

After decoding a DTMF tone, a way of opening and closing the appropriate switches to control the different electrical appliances has to be found. Also the current that is allowed to pass through the switches has to be determined. For example a cooker consumes more power than a television set, so it needs a more powerful switch to switch it on and off.

It is good the controller to be user friendly, so a display and/or indicators should be added to notify the user for the different states of the device. Again the cost should be kept as low as possible. The amount of efficiency of the device is determined by the cost and the features added to it. There has to be a balance between these two. This means that the controller has to have many features and the cost of producing it should be the lowest possible for these features. The extra features that can be added to the controller have to be thought too.

After a lot of research in the World Wide Web and in other resources and continuous work into the design of the controller, a final design was produced. On figure 11 below the main block diagram of the controller can be seen. The signals flow will be discussed first and then more details for each block will be presented.

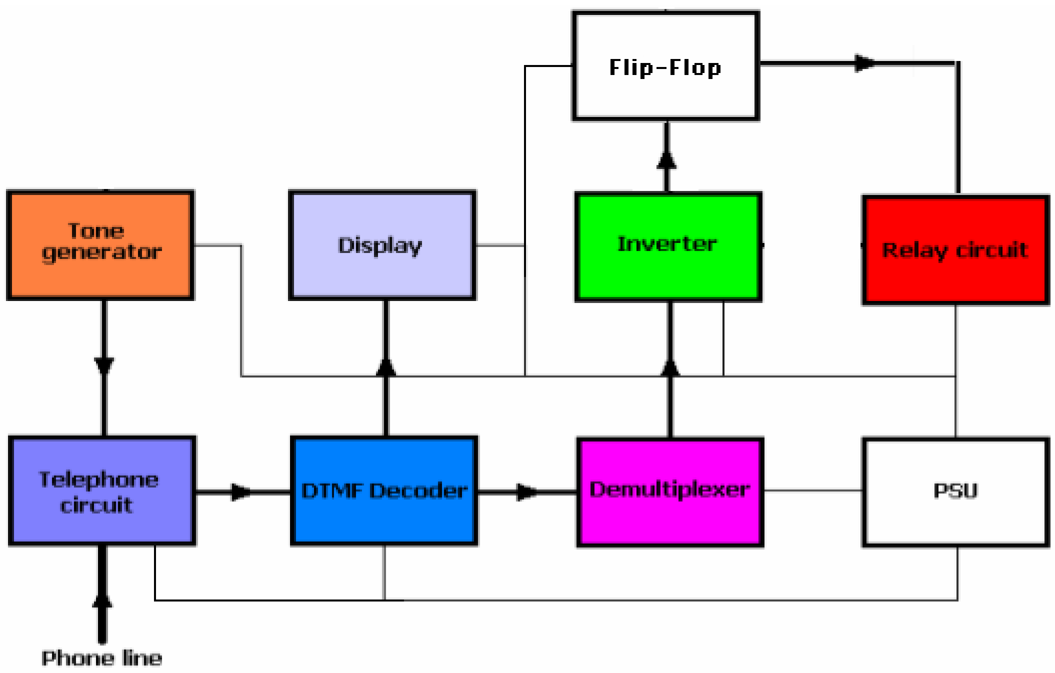


Figure 11. Main block diagram of the controller.

In the above diagram the bold lines represent the signal flow and the thin ones the power supplied to the blocks. The signal from the telephone line passes through the telephone circuit. The telephone circuit converts the electrical signals into audio tones and sends them to the DTMF decoder. The DTMF decoder decodes these tones into a series of bits and outputs a bit combination to the demultiplexer and the display circuit. The display circuit (driver and display) “translates” the combination of bits coming out of the DTMF decoder into the relevant digits and these are displayed to the user. The demultiplexer converts the combination of bits coming out from the DTMF decoder into discreet outputs and drives the inverter. The inverter inverts these discreet bits into their original values (the role of the inverter will be discussed later on) and drives the flip flop circuit. The flip flop circuit behaves like a buffer and keeps the state of the current output when another output is triggered. The output of each flip flop drives the relay circuit to trigger the relevant relay. Depended on the number of output bit each time, the relay circuit triggers the relevant relay and switches on or off the appropriate device connected to it. The telephone circuit converts voltage levels into audio tones (as mentioned above) to drive the decoder and auto answers the user calls. The tone generator inserts an audio tone into the telephone line (through the telephone circuit) and notifies the remote user about the mode of operation of the controller. The role of the tone generator and telephone circuit will be discussed later on. All the blocks are connected with the power supply unit. Let us now discuss each block of the circuit in detail.

Telephone circuit

The telephone circuit is the interface that connects the DTMF decoder to the telephone line. In fact this circuit is tapped to the telephone line. The DTMF tones that the user sends through the telephone line can reach the remote telephone and the controller at the same time. The telephone circuit uses capacitors to block the telephone line voltage and to allow only the audio signal to reach the DTMF decoder input. Two modes of operation can be used; the single ended input mode and the differential input mode.

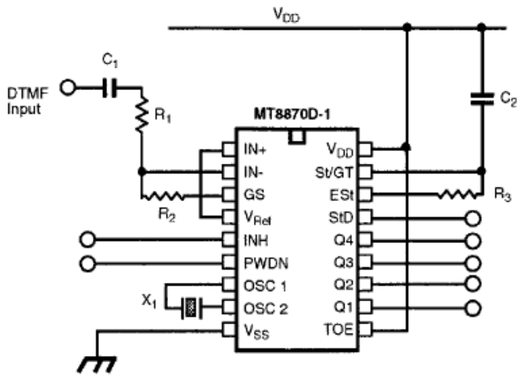


Figure 12. Single ended input mode.

In the single ended input mode one of the two wires is connected to the capacitor. This could give some problems and in fact it gave problems when testing the circuit. The controller did not respond correctly to the user input, so the differential input mode was selected.

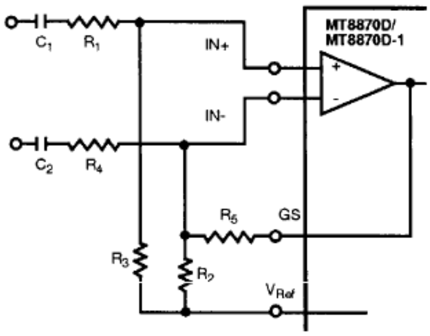


Figure 13. Differential input mode.

In the differential input mode both wires of the telephone line are connected to capacitors. The audio signal then is fed to a differential amplifier inside the DTMF decoder chip.

The telephone circuit also, auto answers the call that the user makes and allows the tone generator to insert a tone into the telephone line to notify the remote user that the line is in controller mode. This was the final decision for the telephone circuit. The first approach was to have the remote user remotely select between the telephone mode and controller mode. Based on this first approach after the user listens to the first “diiiiiit” sound when making a call to the controller, he should press “*” to activate the controller (answering the line by connecting a 220Ohm resistor across the telephone wires), and then he should hear a tone to inform him that the line is in controller mode. After testing the circuit it was found that before the line was answered, it did not pass any audio tones into the DTMF decoder. Thus the controller could not connect the 220Ohm resistor and so the user could not make it answer the line. In order to find what the problem was, a set of headphones were connected after the two capacitors of the telephone circuit to listen to the audio tones that are coming out of them. When a call was made to the controller, every time a “diiiiiit” sound was happening on the calling phone, a “crack” sound could be heard into the headphones in the controller side, so there was an assumption that a signal was passing through, but no tone could be heard. Connecting a voltage meter to the telephone line could show what was happening, but this is prohibited by the telephone companies. The circuit was changed back to the single input mode, hoping that this will solve the problem, but it was not. In order to solve the problem of auto answering the call, a circuit was designed to detect the ringing and automatically close a relay switch. Then this circuit could be used to connect the 220K resistor to the telephone lines and answer the line each time a call is made to the controller. This could solve the problem of auto answering the call but it also has a disadvantage. The remote user cannot switch between controller mode and telephone mode remotely. But does he really need to? Imagine the case scenario that the user leaves home (so there is no one left home) and he switches on the controller manually (controller mode). He then can call home and switch on and off the appropriate devices. If someone else was left home, he could do that for him and there was no need for the controller to do so. When the user returns back home he switches off the controller and the telephone mode is now on. Besides, using the auto answering has another advantage. The number of devices that can be used is increased by one, because the "*" can now be used as an additional switch and not as a button to switch between the telephone mode and the controller mode.

So this is how this system works. The user leaves out home and activates the controller (a flashing green light notifies the user that the controller is on). The telephone line and the telephone device are connected to the controller. Then the user can call home. The controller detects the ringing and toggles a relay switch. This is a double relay switch. The one contact toggles a 220Ohm resistor across the telephone line in order to answer the call. The other contact toggles (connects and disconnects) the audio tone circuit to the telephone line, in order to insert a tone into it for a few seconds and notify the remote user that the controller is activated, i.e. report the status of the line. The 220Ohm resistor is connected and disconnected after a few seconds (toggling). This is like answering the call and then hanging down immediately. If the remote user that has made the call hasn't hang down, even if the local user has done that already, (this is the case here) then the line stays on and it can be used for 30 seconds before the telephone company hung it down automatically after this period (note, this is tested successfully with Greek providers). In this period of 30 seconds the remote user can press different keys in the telephone keypad (each one followed by "#") to trigger the relevant relays. The role of "#" will be described later on. After 30 seconds the line closes automatically. If the remote user needs more time, he has to make another call. If the remote user needs less than the available time, he can hang down and close the line himself, before the period of 30 seconds expire. At any time, if a local user hangs up the telephone he can use the controller to switch on/off the appropriate devices.

DTMF decoder

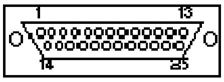
The role of the DTMF decoder is to decode the DTMF audio signal coming out from the telephone circuit into a combination of bits. The solution of software in a microcontroller was avoided because it would be more expensive, it would take more time to write the software and even then it could contain some bugs. The ready made MT8870 DTMF decoder chip was selected because it is relatively cheap and it is widely available in the market today. This chip receives audio DTMF tones in its input and outputs a combination of four bits, dependent on the DTMF tone. This sequence of bits is used to drive the demultiplexer. On figure 14 the decode table for this chip can be shown. The Qx letters represent the outputs of the decoder.

Digit	TOE	INH	ESt	Q_4	Q_3	Q_2	Q_1
ANY	L	X	H	Z	Z	Z	Z
1	H	X	H	0	0	0	1
2	H	X	H	0	0	1	0
3	H	X	H	0	0	1	1
4	H	X	H	0	1	0	0
5	H	X	H	0	1	0	1
6	H	X	H	0	1	1	0
7	H	X	H	0	1	1	1
8	H	X	H	1	0	0	0
9	H	X	H	1	0	0	1
0	H	X	H	1	0	1	0
*	H	X	H	1	0	1	1
#	H	X	H	1	1	0	0
A	H	L	H	1	1	0	1
B	H	L	H	1	1	1	0
C	H	L	H	1	1	1	1
D	H	L	H	0	0	0	0
A	H	H	L	undetected, the output code will remain the same as the previous detected code			
B	H	H	L				
C	H	H	L				
D	H	H	L				

Figure 14. Decode table.

Data port

An important feature of the controller is the data port. It allows the user to control the controller by using a personal computer and a piece of software. The easiest way to do this is to use the parallel port of the computer to send directly the combination of four bits to the controller. The pin schematic for the parallel port of a personal computer is shown on figure 15.



View is looking at
Connector side of
DB-25 Male Connector.

Pin	Description	
1	Strobe	PC Output
2	Data 0	PC Output
3	Data 1	PC Output
4	Data 2	PC Output
5	Data 3	PC Output
6	Data 4	PC Output
7	Data 5	PC Output
8	Data 6	PC Output
9	Data 7	PC Output
10	ACK	PC Input
11	Busy	PC Input
12	Paper Empty	PC Input
13	Select	PC Input
14	Auto Feed	PC Output
15	Error	PC Input
16	Initialize Printer	PC Output
17	Select Input	PC Output

Pin Assignments

Note: 8 Data Outputs
4 Misc Other Outputs

5 Data Inputs

Note: Pins 18-25 are
Ground

Figure 15. Parallel port of the computer.

Confusion could be caused if the user tries to control the controller using both the parallel port and the telephone at the same time. In order to avoid this, a switch has been added to the circuit that disables the DTMF decoder (switches off the power) when the controller needs to be controlled by the parallel port (data). By setting this switch to data mode the controller receives data from the computer only.

Demultiplexer and Inverter

A demultiplexer is used to take the combination of bits coming out from the DTMF decoder and “translate” it into discreet bits. Since MT8870 outputs a combination of 4 bits, a 4 to 16 demultiplexer (4^2=16) is needed. This is because with 4 bits 16 combinations can be made. The 74LS154 is a common 4 to 16 demultiplexer and its outputs are inverted. On figure 16 the function table for this demultiplexer is shown. Since the telephone keypad has 12 digits, only the dashed areas are used.

Inputs							Outputs															
G1		G2	D	C	B	A	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
L		L	L	L	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	**1**	L	L	L	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	**2**	L	L	L	H	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H
L	**3**	L	L	L	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H
L	**4**	L	L	H	L	L	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H
L	**5**	L	L	H	L	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H
L	**6**	L	L	H	H	L	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H
L	**7**	L	L	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H
L	**8**	L	H	L	L	L	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H
L	**9**	L	H	L	L	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H
L	**0**	L	H	L	H	L	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H
L	*****	L	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H
L	**#**	L	H	H	L	L	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H
L		L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H
L		L	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H
L		L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L
L		H	X	X	X	X	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
H		L	X	X	X	X	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
H		H	X	X	X	X	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H

H = HIGH Level
L = Low Level
X = Don't Care

Figure 16. Function table of the 74LS154 demultiplexer.

The fact that the demultiplexer has inverted outputs means that when an output is logic 0, this actually represents logic 1. To deal with this situation an inverter has to be used in each of the outputs of the demultiplexer so that the correct logic is taken out of it. The CD4049 inverters chip was used. It contains 6 inverters into a single package. Since the numbers on a telephone keypad are 12, 2 inverter chips have to be used (note only 12 of the 16 outputs of the demultiplexer are actually used).

Flip-Flop

When the circuit was tested it was found that the solution of the demultiplexer and the inverters was not enough. The status of a switch was not kept when another one was triggered. A way has to be found to keep the status of the switches. A research was done for schematics on the web and magazines and useful advice was taken from an electronic engineer and it was found that a D-type flip flop could be used at each of the outputs of the inverter chips.

Each output connects to the clock of the flip flop. Based on the truth table of the flip flop on figure 17 whenever a clock pulse (5V i.e. a bit (rising edge)) comes to the clock input of the flip flop the "Q" output of the flip flop is triggered. Whenever a 0V (i.e. no bit (falling edge)) comes to the clock input of the flip flop the "Q" output of the flip flop remains where it was before (i.e. not triggered).

That seems to solve the problem of keeping the status of the switches. In fact it gives another problem. Imagine that the user presses “9” button to trigger the 9th switch. If 9th switch previous state is off (i.e. 0V at the 9^{th} output of the demultiplexer output) now a clock pulse will arrive at the flip flop (i.e. 5V (rising edge) at the 9th output of the demultiplexer output) and the 9th switch will be switched on. The problem arises when the user wants to press again directly the 9 button, without pressing any other button between these two moves (i.e. pressing 9 to switch the switch on and then directly 9 to switch it off). In that case, when first pressing “9” button, the 9th output of the demultiplexer will be 5V. This means a rising edge for the flip flop and the flip flop is triggered. When the user presses directly again the 9 button, the 9th output of the demultiplexer will stay at 5V, as it was before, so there is no rising edge for the flip flop to be triggered.

Someone had to deal with that situation or else the controller wouldn't have the expected behaviour. The outputs of the demultiplexer had to somehow changed directly after each command (each press of a button) in order to create something like a reset for the demultiplexer (i.e. to create a falling edge in each output, which would not affect the outputs of the flip flops because they are triggered only by rising edge pulses). Then the user could even press again the same button and create another rising edge, independent of the previous state of the output, which would then trigger the flip flop.

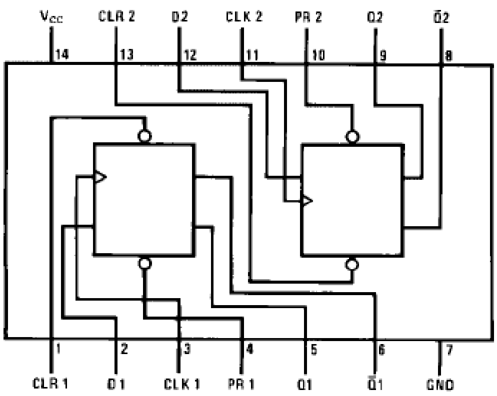
It was decided to have the user pressing the "#" button after pressing each number of the switch he wants to rigger. So he should press for example 9# to trigger the 9th switch. The "#" behaves as a reset, as explained above, and ensures that the relevant switch will be triggered.

The use of "#" after each command has some disadvantages and advantages. The disadvantage is that the number of devices that can be controlled by the controller is reduced by 1. The advantage is that except the reset function there is another switch available to be used for other functions for future development (“#” button also can trigger a relay). Another advantage is that it makes the controller friendlier. The user knows that each digit is a switch and the "#" button is used for other purposes.

For the implementation the first thing that had to be done was to search for flip flop chips that contain more than one flip flop in the same package, to reduce cost and space on the board. Each flip flop in the chip should have a separate clock. Besides, each flip flop should be triggered in the rising edge of its clock only (i.e. only when a bit arrives).

The 7474 chip meets these requirements. It is a chip that contains two flip flops in the same package, using independent clocks for each one. Since there are 12 keypad buttons only six of these chips are needed. This device can be used for shift register applications, and, by connecting Q output to the data input, for counter and toggle applications. The logic level present at the D input is transferred to the Q output during the positive going transition of the clock pulse. Setting or resetting is independent of the clock and is accomplished by a high level on the set or reset line, respectively. On figure 17 the connection diagram and the truth table for this flip flop can be seen.

Connection Diagram



Function Table

Inputs				Outputs	
PR	CLR	CLK	D	Q	$\overline{Q}$
L	H	X	X	H	L
H	L	X	X	L	H
L	L	X	X	H (Note 1)	H (Note 1)
H	H	↑	H	H	L
H	H	↑	L	L	H
H	H	L	X	Q_0	$\overline{Q_0}$

H = HIGH Logic Level
X = Either LOW or HIGH Logic Level
L = LOW Logic Level
↑ = Positive-going transition of the clock.
Q_0 = The output logic level of Q before the indicated input conditions were established.

Note 1: This configuration is nonstable; that is, it will not persist when either the preset and/or clear inputs return to their inactive (HIGH) level.

Figure 17. 7474 chip connection diagram and function table.

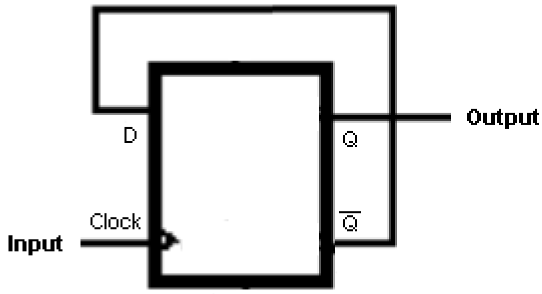


Figure 18. Status keeping solution.

Figure 18 represents the final solution as far as concern the keeping of status of the switches using the flip flop. If someone looks at this picture and the truth table of the flip flop he can see that when the user presses the relevant key there is a rising edge at the clock input. Let's assume that the Q bar at the beginning is 1 (therefore D becomes 1). When the rising edge of the clock happens and D is 1, then Q goes to 1 and Q bar goes to 0. Since Q now is 1, the switch has been turned on. Then the user presses #. This sets a falling edge of the clock (behaves like a reset) which does not affect Q and Q bar, independent of the D. So Q does not change and the status of the switch does not change. When the user presses again the same key, there is another rising edge of the clock. From the previous state, Q bar has become 0. This sets D do be 0 now. When the rising edge of the clock happens and D is 0, Q goes to 0 and Q bar goes to 1. Since Q now is 0, the switch has been turned off. This cycle repeats for every switch and the status of the switches is kept as is, and it is changed only when the relevant key is pressed.

Relay circuit

Each bit (0-9 and "*") out from the inverters goes into a relay circuit to trigger a relay switch. Since star "#" is used for reset purposes (as described above), only 11 similar relay circuits are needed, one in each output of the inverters.

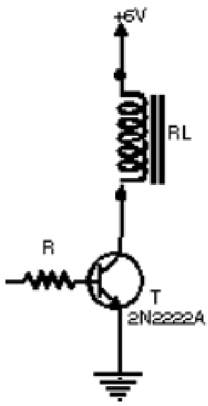


Figure 19. Relay circuit.

In the above figure the relay circuit can be seen. It accepts bits (5 Volts) out from the inverters and triggers the relay. When a bit arrives, its voltage is reduced by the resistor to drive the transistor. A voltage in the base of the transistor makes it behave like a closed switch and current is allowed to pass through it. Current passes from the VCC through the coil of the relay and through the closed transistor to the earth. This makes the relay to be triggered.

During the implementation process it was decided to use double relays in the outputs. The use of double relays has significant advantages. At the start it was thought to use one of the two contacts of each relay in order to activate a LED and notify the local user that this relay *contact* is on/off. Then the other contact could be used for the device connected to the controller (load). But a better solution was found. The LED was connected to the coil of the relay and now the LED reports if the *relay coil* is triggered or not (not the contact). This has many advantages. First of all both contacts of the relay can be used to activate/deactivate devices. For example someone can connect 11 devices on the controller or 22 devices in pairs! Or he can do his own combinations. He can set one device in one contact of the relay to be switched on while at the same time the other device connected in the second contact to be switched off. For example he can switch off the internal lights at his house while at the same time he switches on the garden lights. The combinations that can be done are unlimited. The user can also connect a high current consuming device to both contacts of the relay. With just one contact this would be impossible because the current of this device would exceed the nominal value of just one relay contact. By using both relay contacts he can do that.

Tone generator

The role of the tone generator is to generate a continuous audio tone and insert it into the telephone line for a few seconds (through the telephone circuit). This notifies the remote user that the line is in controller mode.

As a tone generator a cheap 555 timer can be used. On figure 20 there are two basic 555 square wave oscillators used to produce a 1 KHz tone from an 8 ohm speaker. These can be used as tone generator circuits. In the circuit on the left, the speaker is isolated from the oscillator by the NPN medium power transistor which also provides more current than can be obtained directly from the 555 (limit = 200 mA). A small capacitor is used at the transistor base to slow the switching times which reduces the inductive voltage produced by the speaker. Frequency is about 1.44/(R1 + 2*R2)C where R1 (1K) is much smaller than R2 (6.2K) to produce a near square wave. Lower frequencies can be obtained by increasing the 6.2K value, higher frequencies will probably require a smaller capacitor as R1 cannot be reduced much below 1K. Lower volume levels can be obtained by adding a small resistor in series with the speaker (10-100 ohms). In the circuit on the right, which was used, the speaker is directly driven from the 555 timer output. The series capacitor (100 uF) increases the output by supplying an AC current to the speaker and driving it in both directions rather than just a pulsating DC current which would be the case without the capacitor. The 51 ohm resistor limits the current to less than 200 mA to prevent overloading the timer output at 9 volts. At 4.5 volts, a smaller resistor can be used. The tone generated by this circuit is directly injected into the telephone wires through the telephone circuit.

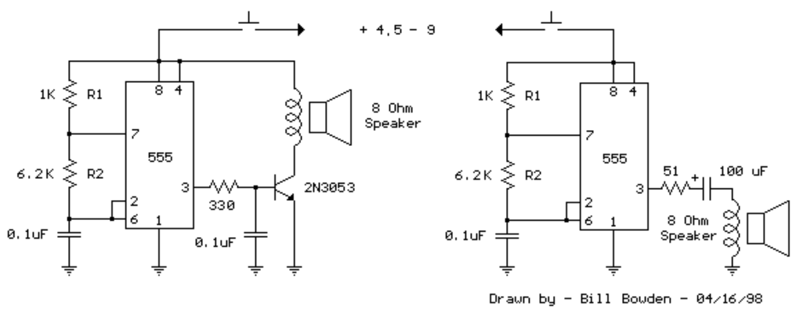


Figure 20. Two tone generator circuits.

Display

As specified in the requirements the controller should be friendly to the user. A display has been added for this reason to notify the user what switch has been triggered. The display has also been added for debugging purposes during the implementation process.

As far as concern the display choice, a seven segment display was selected. Seven segment displays are cheap compared with other types of displays and they are reliable. Besides a seven segment display consumes much less power (maximum seven LEDs operating) and it can be very easily driven by using a 74LS47 driver chip. This chip takes directly the combination of four bits coming out of the DTMF decoder and switches on the relevant segments in the seven segment display in order to construct the relevant digit each time. A common anode display was selected and each cathode was connected to the driver chip through a resistor.

Power supply unit

The power supply unit is connected to all the active devices and it should provide enough power to these. Besides it should be cost effective. The best solution is to have a wall socket un-stabilized power supply unit which is connected to the controller with a long cable. In that way high voltages is kept off from the controller and the interference caused by the 50Hz AC is the minimum possible. This has also another advantage. It allows driving the controller from a low voltage source (where high AC voltage is not available) by just removing the wall socket and connecting the controller directly to the low voltage.

At the controller side there is a 7805 voltage regulator, which stabilizes the low voltage and regulates it to the wanted value (5 Volts). This regulator can handle up to 1Amp current, which is much more than it is needed by the controller. Since a voltage of 5V is only needed the regulator was connected as shown on figure 21. Although no output capacitor is needed for stability, it does improve transient response. The input capacitor is required if the regulator is locate an appreciable distance from the power supply filter.

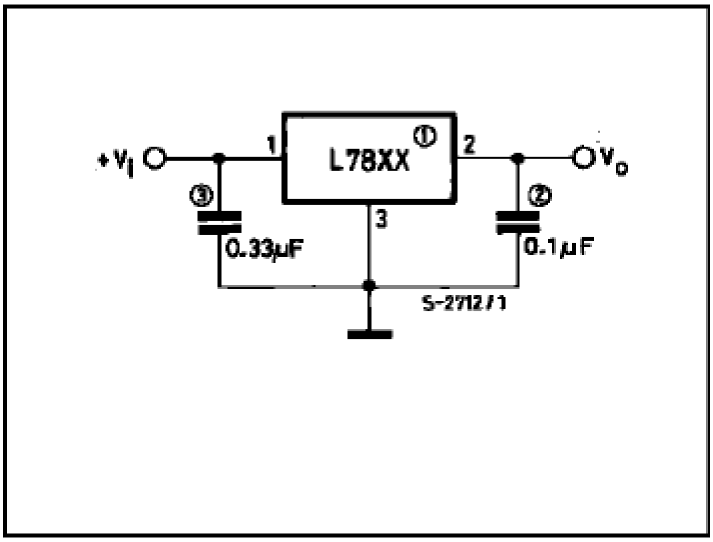


Figure 21. Regulator connection diagram.

Computer program

As an option it was asked to be able to control the controller by using computer software. It was asked either to write from scratch computer software, or to use an existing one for this purpose.

The data port on the controller is a parallel port and accepts a combination of four bits as mentioned in previous section. These four bits are the same that the DTMF decoder sends to the demultiplexer when operating in DTMF mode. All that is now needed is to have software that will be able to send this combination of four bits through the parallel port of the computer to the demultiplexer inputs.

Jem E. Berkes has produced a wonderful piece of software that fully controls and debugs the parallel port of the computer. The purpose was not only to have software to trigger the relevant outputs but also to debug the parallel port in order to prove that the controller has the expected behaviour. So it was decided to use this software to control the device from the computer. By using it someone can manually set the combination of four bits to send through the parallel port. A screenshot of the running program can be seen on figure 22. The user can chose the parallel port mode by using the left and right arrows, select the bit to change by using the up and down arrows and set or reset this bit by using the enter key.

The drawback of using this software is that the user can only change one bit each time. If he wants to change two bits he has to make a second movement. But the first movement can accidentally trigger another switch. Another thing is that he has to manually reset the flip flops each time from the program (like the "#" when operating in

DTMF mode). To illustrate how the program works imagine the initial combination of bits is 0000. Then all the switches are open (off). If the user sets the first bit, this will give a combination of 1000 coming out from the parallel port to the controller, thus the first switch will be triggered. Then the user has to reset this bit to perform the reset function for the flip flops, so he has to send a 0000 combination again to the controller. If the user wants to trigger the third switch he has to send 1100. But he can only do that in two steps. First he will send 1000 to set the first bit and then 1100 to set the second too. Then the third switch will be triggered. But the first step (1000) will also trigger the first switch. Also, the reset function is performed in two steps. The user can reset one bit at a time. This of course can give problems when using the software for commercial applications but as explained above the purpose of using this package is to debug the controller and prove that it works rather than creating a commercial application.

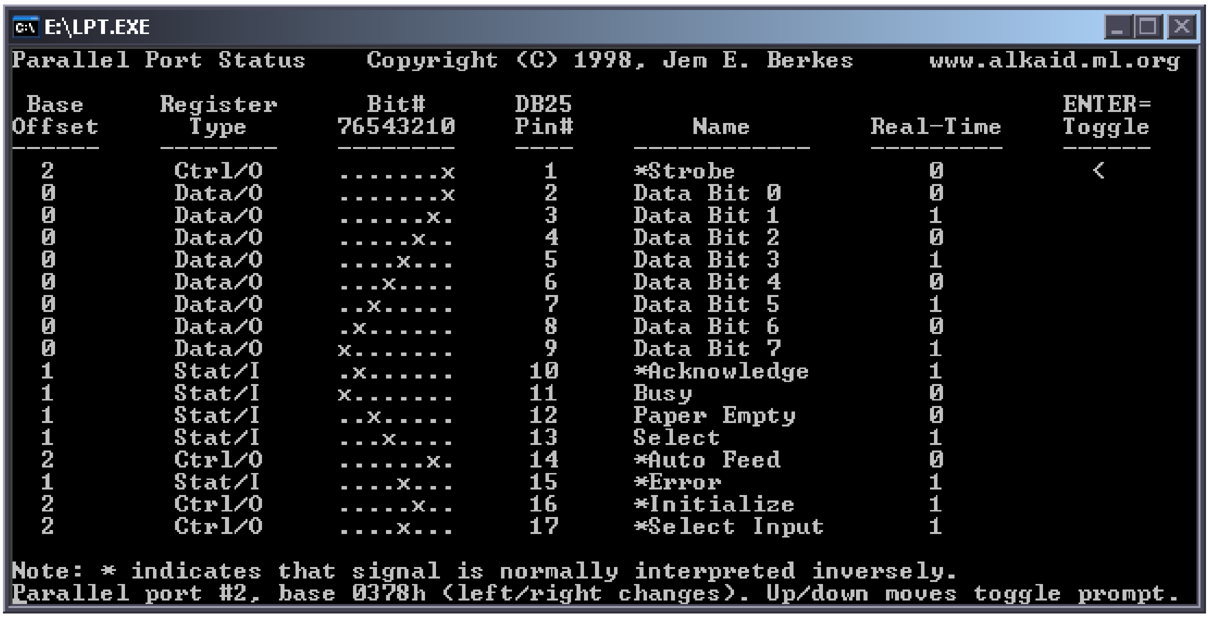


Figure 22. Computer program.

An important thing has to be mentioned at this point. Windows XP and Windows 2000 operating systems have the parallel and serial ports of the PC locked in order to protect them from intruders. So these ports can not be directly accessed from the programs. Finding drivers to control them is not too easy and when finding one, someone has to co-op with the specific features of the driver. That is why it was decided to use the DOS operating system for the program to run. A bootable DOS CDrom has been made which include the program inside. This CDrom can be run from Windows too. If the Windows used operating system is other than Windows XP and 2000 then the user can insert the cd, double click the LPT.exe and run the program. In any other case he has to follow the next few steps to run the program. He has to set its computer (from the BIOS) to boot from CDrom first. Then he must insert the cd into the CDrom and boot his computer. To run the program he must type "LPT" at the command prompt and he can access directly the parallel port of the computer.

Implementation

The controller was implemented based on the research work that had been done. Although it was implemented on a breadboard, a PCB schematic was produced using the Express PCB design package. This allows a professional reproduction of the controller to be done in the future. Knowing that it is difficult to find PCBs larger than A4 size, it was decided to break the main PCB into two smaller ones. The two PCBs can be connected together using a band of cables. They can also be folded, the one above the other, to fit in a smaller box. A circuit schematic was also produced using the same package.

The implementation on the breadboard began during the research. Many things were changing until the final solution was found. The breadboard is the most flexible in changing and testing different circuits. Although the controller supports eleven devices (or twenty two in pairs) only three flip flop and relay circuits were constructed. This was mostly done because these relays are costly devices. Using eleven relays (plus one for the auto answering circuit) would cost too much, which was not afforded. The operation of the controller can be seen by using just two or three relays. The other relays have exactly the same behaviour.

An important notice has to be made here. The controller was tested and worked perfectly fine using non-UK telephone providers. It might not have the expected behaviour using UK telephone providers. The remote operation of the controller is based on the fact that a provider would let the user use the telephone line for a few seconds before it hung it down automatically.

Conclusions

Here the conclusions from the overall work are presented. The requirements of the project were met and an intelligent and cost effective controller was produced. The cost for this controller was kept as low as possible without compromising the performance of the system. Computer software was used in order to control the device through a personal computer. The overall work has been done based on the timetable as much as possible. Many times it was needed to change the preliminary design and reconstruct the controller until a satisfactory solution was found.

The project was proved to be very useful because it was a chance to learn how it is to undertake such a large amount of work and to work professionally to finish it. It gives the essential experience to the developer which will help him in the future to get a job. Plenty of time was spent on this project but the final result was satisfactory because the controller design and implementation was finished. This project is a proof of the work that has been done and it is a metric of the capabilities and experience of the developer in his future career.

Recommendations

The controller has some very important features that approach a professional solution. Despite that, if a company decides to produce a commercial version of the controller in the future it can be made more advanced and easy to use.

An important option of reporting the status of each individual contact inside each relay to the remote user can be added. Then the user will know if the device is switched on or off. Another option could be the voice status reporting. Instead of using a simple and cheap tone generator to report the status the company could use a voice recorder/reproducer chip. This device will give instructions ("talk") to the remote user instead of just generating tones. This will make the controller friendlier to the user. A very important feature which would enhance the security of the controller would be to add a password option. This ensures that only the authorised user could be able to send commands to the controller. The device can also be made mobile. By using a mobile or a cordless phone it can be controlled without the need for telephone lines.

The other thing that could be improved is the software part of the controller. A more advanced software could be written which would allow the user to select directly the number of the switch to trigger or to set up timers for each individual switch, in order this to remain on or off for a predefined period of time. This software could also report the status of these switches back to the computer. It can also have graphical user interface to make it easier to use.

Some recommendations related to cost have to be done at that point. The cost for this project is basically the cost for the hardware components that have been ordered. The PCB designing software that was used is freeware. As mentioned in previous section the cost should be kept as low as possible without compromising the performance of the system. Only the essential options were added in order to minimise the cost. Engineers know that cost is one of the most important factors when producing a device. This specifies if this device will be produced in large quantities by a company or not produced at all. The controller costs about £100 (including the 12 relays). This is the amount that an individual must spend to construct the device with these features. Of course it would be much cheaper for a company to build the hardware rather than an individual. The companies buy components in large quantities and they buy them cheaper than individuals. Generally, a more detailed research by the members of a production company would improve the features of the controller and minimise the device cost.

References

T.H. Tsim (August 25th, 1994) DTMF FAQ - Telephone Tone Dialing chips V1.20
Retrieved 03-2004 from http://margo.student.utwente.nl/el/phone/dtmf.htm

ZyTrax, Inc. (1994 – 2004) Tech Info - DTMF Tones
Retrieved 03-2004 from http://www.zytrax.com/tech/telephony/dtmf_tones.htm

Teletechnics Afield (Mar 09, 2002) DTMF tones
Retrieved 04-2004 from http://www.teletechnics.co.nz/reference/telecom/dtmf.html

ZMITAC. Theory of Logic Circuits
Retrieved 05-2004 from http://bszx.iinf.polsl.gliwice.pl/TLC/tlc_files/ex11.doc

John Bingham (1996) Connecting French phones to German/Austrian/US RJ11 cables
Retrieved 03-2004 from http://www.frauenweb.at/~tina/myhtml/frenchphone.html

Ken Bigelow (1996, 2000-2004) The Seven-Segment LED
Retrieved 06-2004 from http://www.play-ookey.com/digital/experiments/seven_seg_led.html

Lawrence Mayes (2002-2003) A novel electronic latch design
Retrieved 07-2004 from http://graffiti.virgin.net/ljmayes.mal/var/optoltch.htm

Electronics zone. Remote Control Circuits
Retrieved 05-2004 from
http://electronicsworld.tripod.com/remotecontrolsimages/remotecontrolsckt2.html

IEC circuits archive. Telephone Line Based Audio Muting and Light-On Circuit
Retrieved 08-2004 from http://www.geocities.com/IECMaster/circuits_tel/cir_tel003.html

Bernard Sclar. Digital Communications, Fundamentals and applications (2nd edition)
Tarzana, California

Appendix

In this section the schematic diagrams of the controller, the silk layer and the copper layer of the PCB designed, some controller photographs, a screenshot of the controller program, and the controller semiconductor datasheets that have been used are presented.