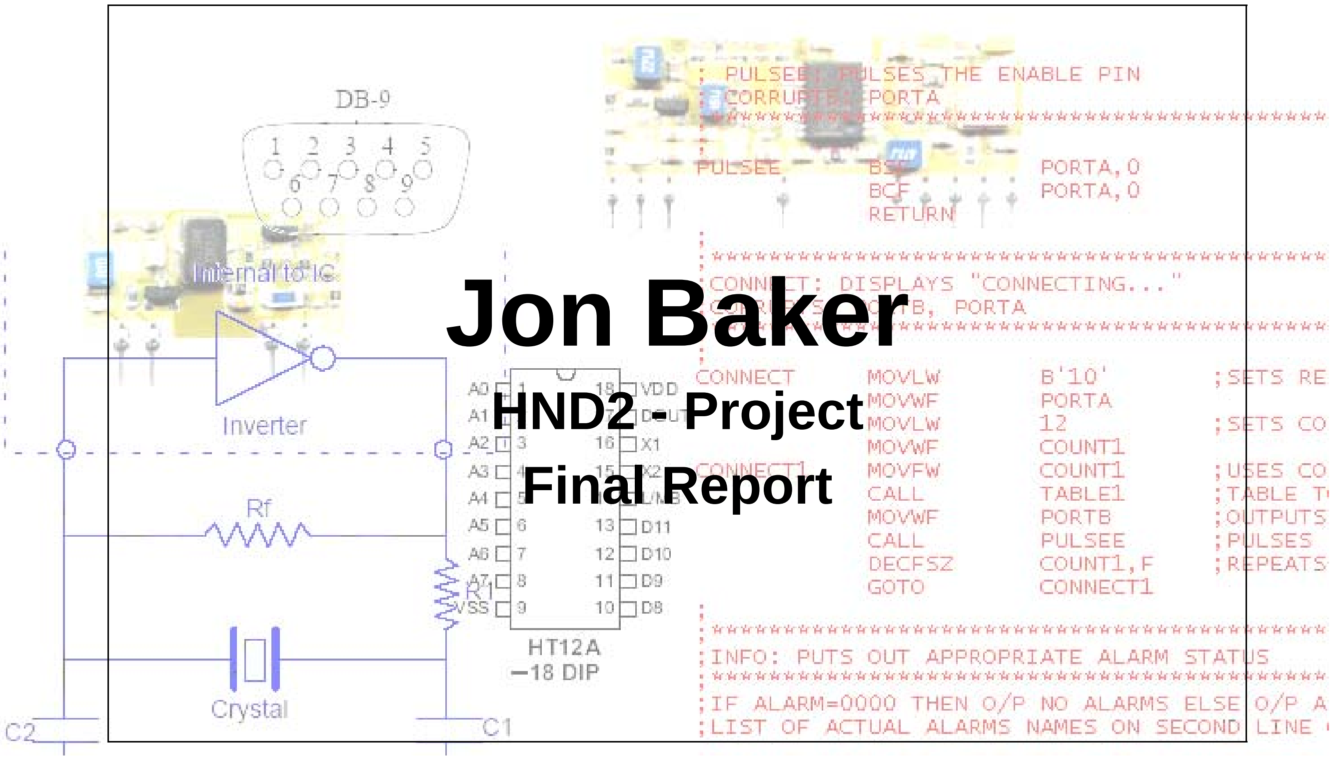


Jon Baker

HND2 - Project

Final Report

Contents

Introduction

Project

For my HND project I have to choose a project to design, research and design it, and then build and test the completed unit.

Initial Design Ideas

When I was starting my project these are the following ideas I came up with along with some advantages and disadvantages:

Wireless Serial Port for PC

- Project I would most like to do – motivation
- Made quite easy using ICs but with scope for a complex project
- Many uses
- Unique to my knowledge
- I'm sure it will expand my knowledge
- Radio circuits are not easy!
- According to my research getting a high bit rate will be difficult
- Could be quite expensive

Graphic Equaliser (LED bars over frequency)

- Again, a project I would very much like to do
- A standalone graphic equaliser is unique to my knowledge
- Would definitely expand my knowledge
- Would be quite difficult to implement
- According to my research the filters involved would be difficult
- Displays could be expensive

Programmable Message Display Board

- Would be a fun project
- Scope for a large range, easy to difficult project
- Could be quite expensive for display

Home Security System

- A useful if common idea
- Again, a large scope of very easy to very difficult designs
- Not that interested in designing
- To do a good design would be expensive

GPS with PC Connection

- Would be very interesting but,
- Very difficult!
- And expensive!

My Chosen Idea

I decided upon the idea of a wireless serial port for a PC, however, I was told that since serial ports are not very much in use now that it would be a good idea to modify the concept. The final idea was that of a wireless alarm system for a PLC. I should explain how this came about and what this means.

A PLC is a Programmable Logic Controller, these are the devices that often will be running production lines in factories. These devices include serial data port(s). I was told a good modification to my initial idea would be to make a device that connects a PLCs serial port to a wireless device for the purpose of being able to know remotely when there is a problem with the PLC itself.

To explain: Imagine a situation where the entire production team is a small group of people. One, two or three people maximum. These people cannot stay around the production line continuously. If someone had to so, this would limit productivity dramatically, however, if someone was able to walk within a reasonable distance of the PLC but still stay informed as to the status of the unit this would be extremely useful.

My First Plan

After researching briefly PLC input/output and considering options about how I could plan this I came up with the following information:

- PLCs are fitted with standard 9/25 pin serial RS-232C ports
- I will need radio transmitter and receiver chips
- These will probably work on 433MHz
- I will need some sort of readout such as an LCD

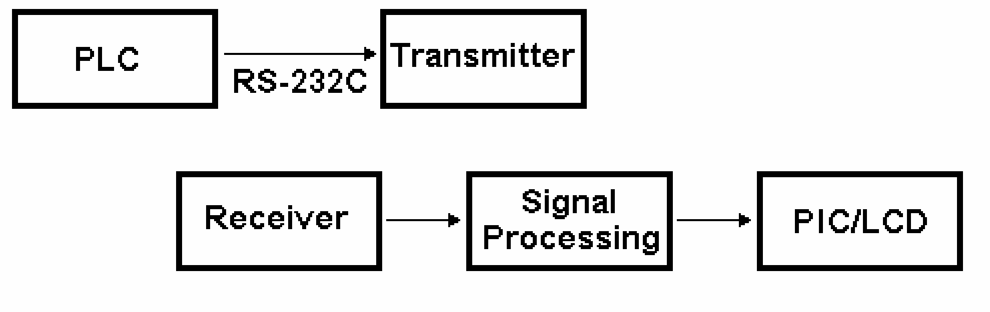
After talking this basic concept through I was given the following recommendation to make the project more complex and interesting:

- User ID Encoding (So that a uniquely identified transmitter could only communicate with the receiver containing the same ID)
- PSK modulation for security
- A system which has four separate alarms
- A PIC and LCD on receiver to allow for a more complex system

From this point I started designing the project.

Design

Basic Outline



This is the basis I started from:

- A connection from the PLC to the transmitter
- A transmitter and receiver pair
- Some sort of processing method
- A PIC running a program to drive an LCD

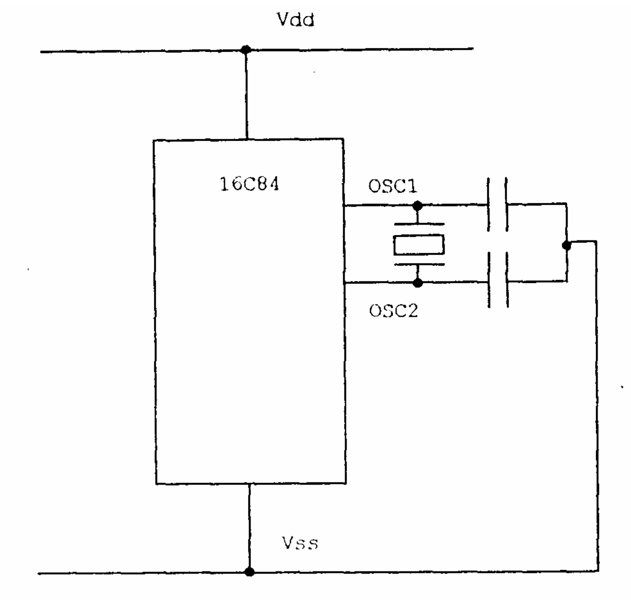
Preliminary Research

A copy of any relevant research gathered is provided in the Appendices at the end of this report. Any internet address or book details are also given in the bibliography.

At this point I am only starting preliminary research with the goal of being able to produce a plan of action for following work and a specification listing the basic parameters of my project.

PIC

I already have experience programming PIC controllers, so using the chip I already know how to program (the PIC16F84) seemed like a sensible idea. I want to use the PIC to drive an intelligent display on my receiver, an LCD. The datasheet for the PIC is available at Microchips website at www.microchip.com/download/lit/pline/picmicro/families/16f8x/30430c.pdf.



Above is the basic circuit diagram for connecting a PIC to work, in this diagram the PIC will simply be able to run free, any other inputs or outputs are not shown.

The following table shows values to be used when connecting the oscillator circuit up:

Mode	**F (Hz)**	**C_1 OSC1 (F)**	**C_2 OSC2 (F)**
LP	32k	68-100p	68-100p
	200k	15-30p	15-30p
XT	100k	8-150p	150-200p
	2M	15-33p	15-33p
	4M	15-33p	15-33p
HS	4M	15-33p	15-33p
	10M	15-47p	15-47p

User ID

The company RF Solutions at www.rfsolutions.co.uk sell chips capable of user encoding. The chips referred to are HT12A and HT12E. However after further research I will not be employing these, see below.

PSK Modulation

As well as this I found in the book *Electronics Communication Techniques* a circuit for 4PSK modulation. I managed to complete the design for this circuit but after pricing up the centre-tapped transformer which alone would cost over £20 I was told to use ICs for the task instead. The following chips will provide the necessary security as well as providing a built in user ID encoding system all on one chip.

1. RF Solutions – www.rfsolutions.co.uk – RF600E/D
2. Maplin – www.rfsolutions.co.uk – N49AU/N50AU

Transmitter & Receiver

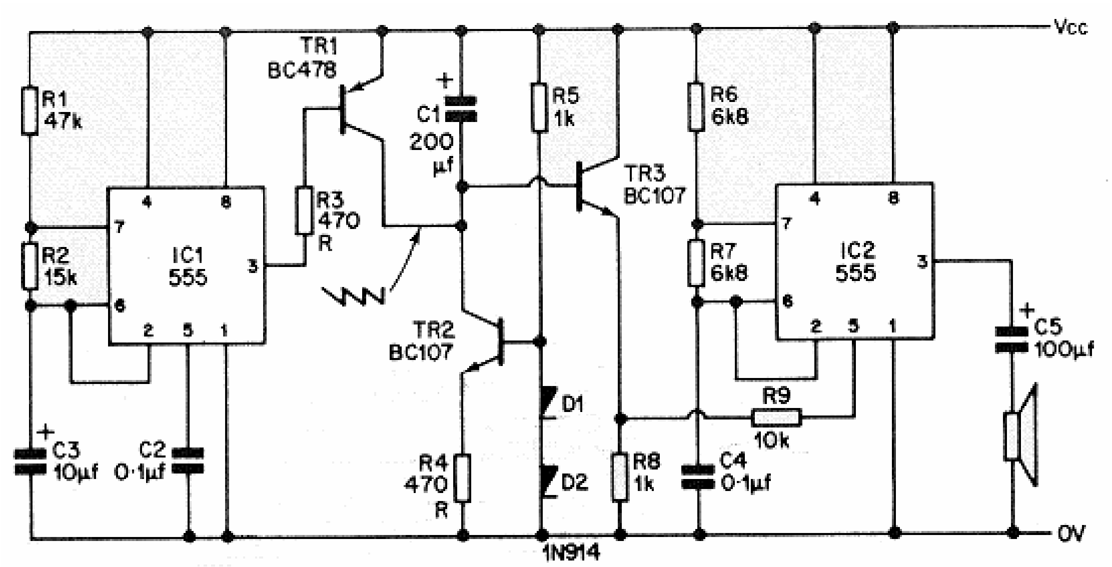
At the time I found a limited supply of transmitter/receiver pairs on the RF Solutions website too as well as some that Andrew used in his previous project from Rentron. Below are the details:

- RF Solutions - www.rfsolutions.co.uk - RTFQ2 and RRFQ2
- Rentron - www.rentron.com - TWS-434 and RWS-434

I was able to get a pair of the Rentron chips and therefore decided to use these.

Alarm

I decided I would need an alarm, some sort of audible or visual alert as well as an LCD readout. I decided I wanted both the alarm and visual alert, a simple LED would do for a visual but I wanted something a little more interesting for an audible alert, something to grab attention. Below is a schematic for a "*Star Trek" Red Alert Siren* taken from the book "555 Projects".



I decided to research PLCs to find what sort of connection was necessary. I found a book on PLCs at http://claymore.engineer.gvsu.edu/~jackh/books/plcs/. The necessary information about the communications ports can be found the in the appendices. PLCs utilise a standard PC COM port connector using the RS-232C standard.

Since all of the circuits I wish to build consist either of programmable chips, radio transmitter/receiver or sound output I was not able to simulate these circuits in a computer package therefore I had to build them on breadboards to test that the design works.

With all of the previous research completed at this point I decided to compile my specification and decide upon a plan of action to follow for the rest of my project.

Specification

General Specification

A remote PLC alarm system with a serial connected transmitter for the PLC working on 433MHz secured with Unique ID coding and PSK encoding. This unit transmits to a configurable, portable, PIG driven receiver unit with LCD readout showing which alarm signals have been sent. The readout can show actual characters so that no codes have to be remembered. The transmitter uses the PLCs RS-232 connector.

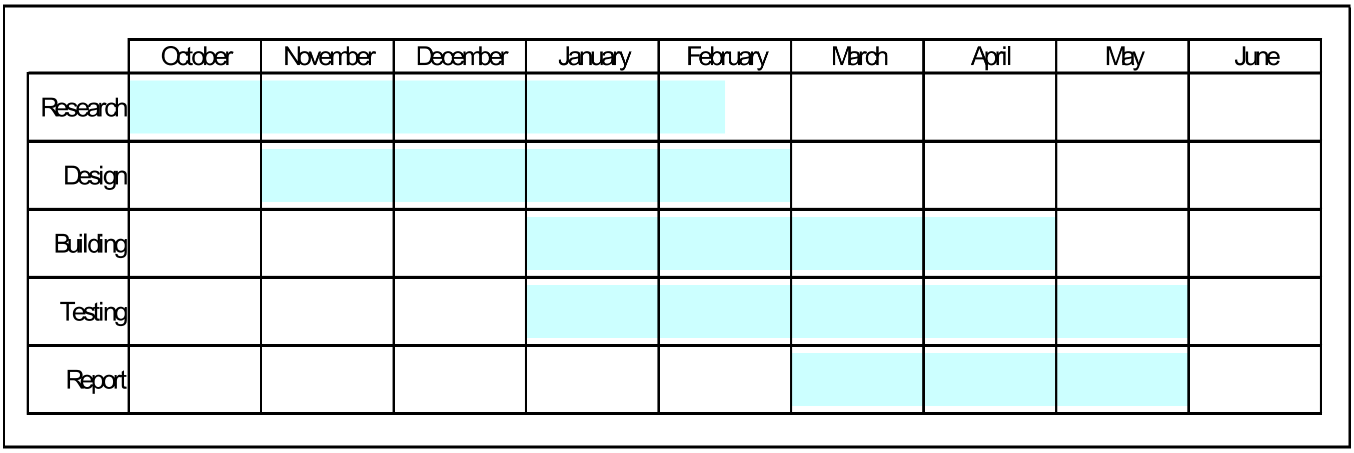
Detailed Specification

Transmitter
Unique Module ID Sending
Internal Hardwired ID selection
AC to DC Adapter Powered
Works on 433MHz
Amplitude Modulation
RS-232 Connector to PLC
Power Switch

Receiver
Battery Powered
AC to DC Adapter Powered
Portable
Works on 433MHz
Amplitude Modulation
Liquid Crystal Display
System controlled via PIC 16F84 Chip
Unique ID Checking
Internal Hardwired ID selection
Power Switch

Plan of Action

Below is the plan of action I decided upon given how well my research had gone up to this point.



Complete Research

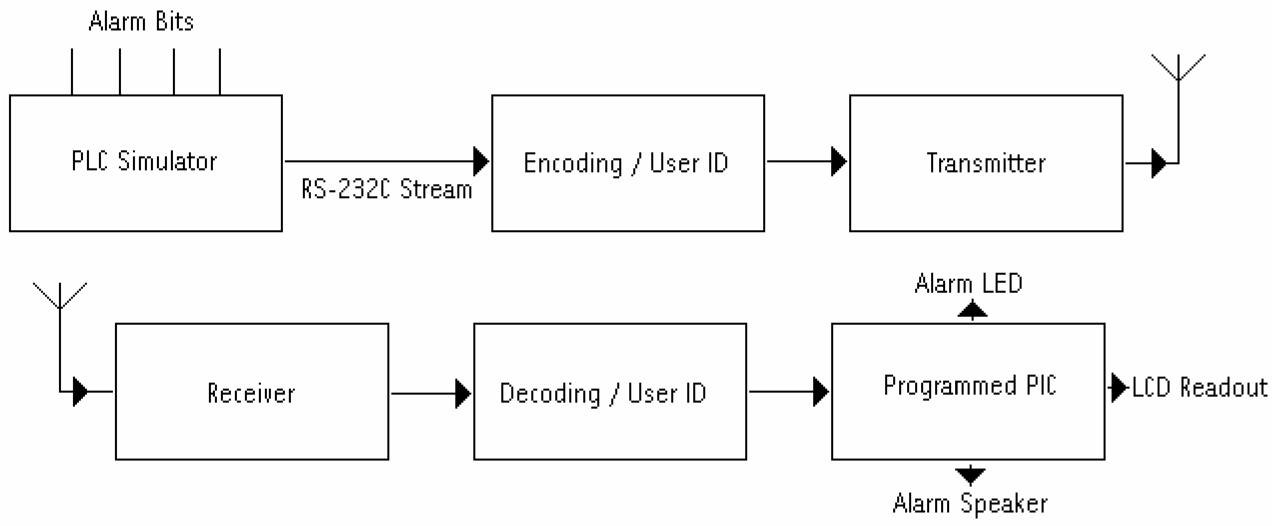
Now that my specification and plan of action is complete the next goal is to complete my designs so that I can produce a component list.

The first important point is that I did not have access to a PLC unit therefore I need to produce some sort of *PLC simulator* that will allow for my project to be tested, since this will not technically be part of my project I did not want to spend too long in the design, testing, and making of this unit. My idea for this unit was a collection of shift registers pre-programmed to output a bitstream to the transmitter along with a number of “alarm bits”. These alarm bits would be changeable by switches on the unit so that I can simulate different alarms on my project. This would compare to the serial output of an actual PLC.

The next section is the transmitter, this simply takes in the bitstream and transmits it. Following this is the receiver, which decodes the incoming signal to the original bitstream.

Finally is the PIC unit itself which will need to be programmed to take the signal, synchronise to it and extract the *alarm bits* then process them and act accordingly.

This is the way in which I have decided my project should function. See the block diagram below for my finished design.



Taking this research now step by step.

PLC Simulator

The PLC Simulator, as I said previously will need to take in four separate alarm statuses and then output this as a bitstream for the encoding transmission stage. This will require a clock, parallel in serial out chips and a format for the data. After considering the synchronisation necessary for the PIC to be able to lock on to I decided that several bits for start and stop (like the serial output of a PC) would be necessary. Below was the format I decided upon:

Start	**Alarms**	**Stop**	
1 0 1 0 1 0 1 0	X X X X	0 0 0 0	Where: X is an unknown alarm bit

I decided upon the start byte because it is much easier to program a repeating pattern for the PIC synchronising, as for the stop bits. The alarm bits will be included by simple switches. This data *frame* is 16 bits long and therefore will require either one 16 bit PISO chip or two 8 bit PISO chips.

I found some appropriate PISO chips in the Rapid Electronics catalogue from www.rapidelectronics.co.uk part 83-0106. They are 8 bit registers. I then designed a circuit to allow for the appropriate configuration of two of these chips along with a clocking circuit. The completed circuit is shown under the *Final Circuits* heading. Although I did not wish to spend too long on this unit as it is technically not part of an actual system just part of my testing I could not avoid putting quite some effort into this as it did require design, testing and building.

Since there are 16 bits to send out the parallel load pins are connected to a divide by 16 circuit to the clock. This means that every 16 clock cycles the ICs will reload their 16 bits every 16 bits, which is what is wanted. The clock is a simple 555 timer. The PISO chips are connected so that the output of one feeds the other thus appearing as a single 16 bit shift register. The output of the most significant IC feeds to the transmitter chip. Each of the 16 bit pins is either connected to +5V or to ground except for the alarm bits which connect to four switches.

Encoding and Decoding

Following the simulator is either going to be an encoding or transmitting stage. I say either simply because of cost. As previously stated building PSK circuits is intolerably expensive, however the cost of inserting encoding and decoding chips is a large percentage of my budget anyway and will therefore be added only after the entire project is completed. My project is designed in a modular fashion and since the security chips work in a simple manner feeding the bitstream through, they can be added at a later date using jumpers.

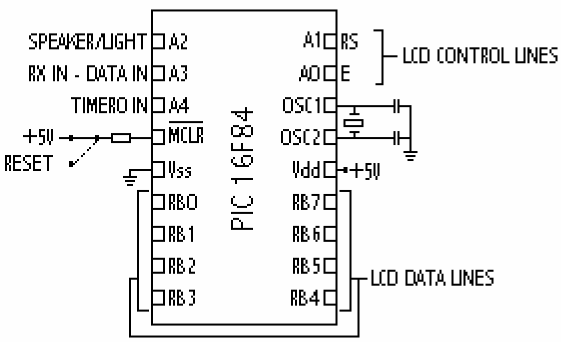
At this time a lack of security chips will be presumed.

Transmitter/Receiver

Using the Rentron chips as previously stated (the RWS-434 and TWS-434) the transmission on reception circuits can be easily built. See under the completed circuits section for the schematics. The only variable here is antenna length, the stated antenna length is 30-35cm. This length will be hard to integrate into the type of device I wish to produce, especially on the receiver. For the moment in testing I am using 10cm antenna.

PIC Receiver

Finally (currently presuming no decoding) the PIC unit. This is the main complex part of the circuit and required much design. I presumed here the digital output of the receiver chip feeds direct into one of the input/output leads on the PIC chip. After this it was necessary to assign input/output leads to everything before proceeding with the programming. Pin assignment is as follows:



Here is the PIC with everything marked. Each and every pin is used, it would have been possible to use the LCD in half-byte mode using only four pins for the data lines but this would have compromised on speed and made the programming much more difficult but there were just enough input/output lines to accomplish the task.

All of port B (RB0-RB7) is used for the LCD data port. The display I use is an alpha-numeric one with its own built in character table. As with ASCII on computers each different character is given an 8 bit code. The function of the display is thus:

- A character code (8 bits) or command is put on to the data lines, PIC port B
- The display is then ‘clocked’ using the control lines connected to port A to either send the character to the displays memory or in other cases execute the command code on port B

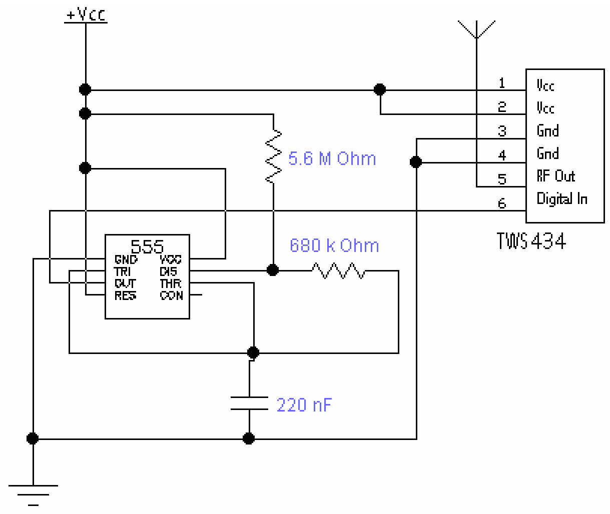
The port A bus is used for multiple purposes. A0 and A1 are used as control lines for the display. Pin A2 is used to activate/deactivate the alarm and LED to indicate alarm(s). Pin A3 is where the receiver digital data out pin is to be connected and finally pin A4 is where the Timer0 input would be. Other circuitry is shown such as the reset circuitry and oscillator.

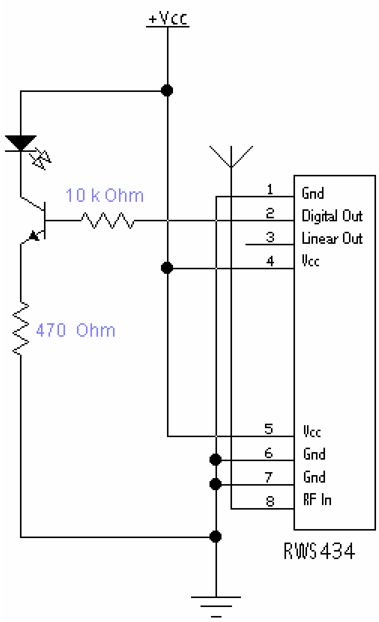
The only other ‘design’ is the PIC software itself. This is without a doubt the most complex part of the project. It requires data synchronisation, data extraction and an LCD interface program. I will not be going into detail here, but the program itself and the flowchart for it can be found in the appendices. Further information can be found in the following section.

Testing the Designs

Transmitter/Receiver

With the circuit design completed testing was next. The first test carried out was on the transmitter/receiver pair. A simple circuit using the TWS434 and RWS434 ICs, a basic 555 clock and receiver LED was built. Shown below:





This test was carried out for two reasons:

1. Simply to test the chips.
2. To test range against antenna length.

The results of this test are given below:

Transmit Aerial (cm)	Receive Aerial (cm)	Distance (m)
8	16	5
16	8	8
8	8	4
4	4	2

This experiment was conducted at 5v and also indicates the transmit antenna has a larger effect on the transmit distance.

The specification lists the following:

- 10cm aerial – 7-9m range
- 30cm aerial – above 25m range (specification says up to 100m)

PLC Simulator

Following this I built the PISO simulator circuit to make sure I would have a working simulator. I built the actual circuit to be put on PCB. Although the actual clock speed was still an unknown the components are the same. The circuit can be found in the Final Circuits section.

The circuit performed as expected when connected as per the datasheet and configured for the stream output as previously specified.

Alarm Circuit

The alarm circuit I took from 555 Projects when built worked correctly. Though I found altering the timing components was necessary to achieve a better sound, testing was conducted as follows:

Components R1, R2, and C3 control the waveform timing at TP1.

The following calculations apply:

Uptime = $0.7(R_1+R_2)C_3$

Downtime = $0.7R_2C_3$

I found that values of R_1=47kΩ, R_2=47kΩ, and C_3=10μF made the most annoying sound.

PIC Receiver

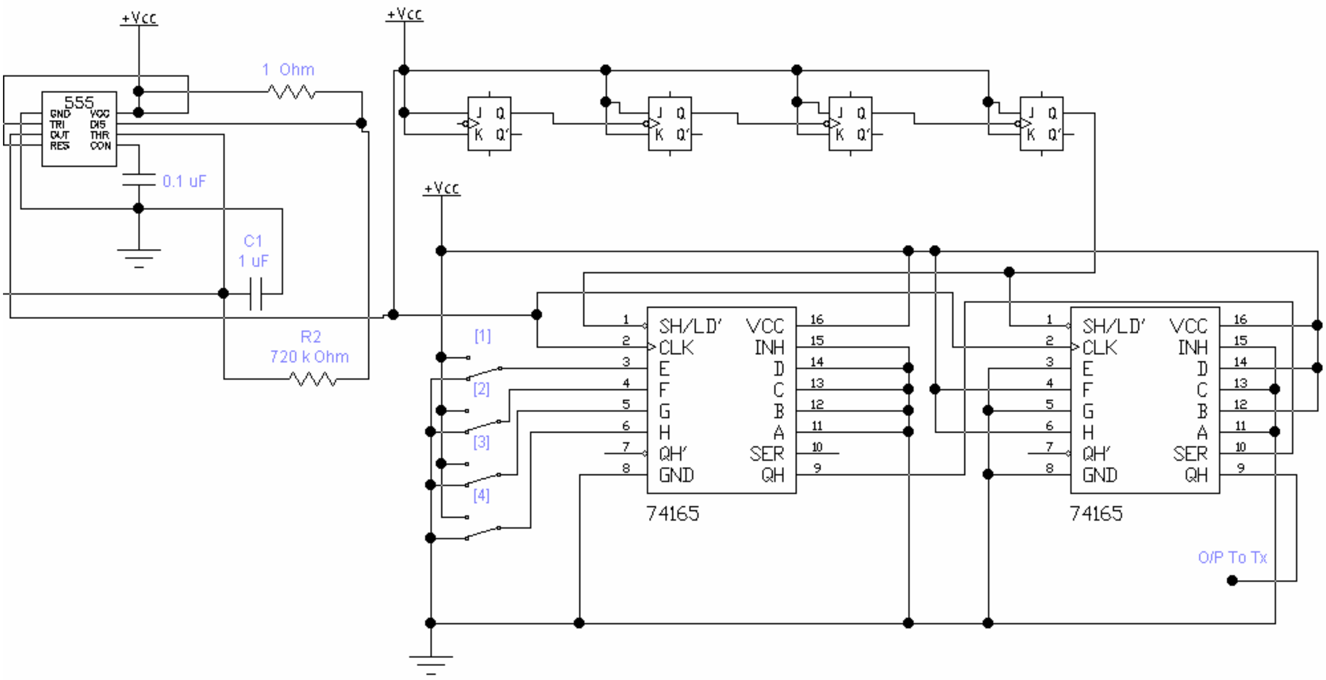
Lastly is the PIC circuit. This cannot be tested without the programming which was not at this stage completed. Though so far the LCD interface was complete as well as message output from pre-programmed alarm statuses.

The only problem so far is synchronisation of bit streams and toggling between the two display lines. Though the programming seems to be correct and all the necessary data lines are connected the second line of the LCD cannot be written to.

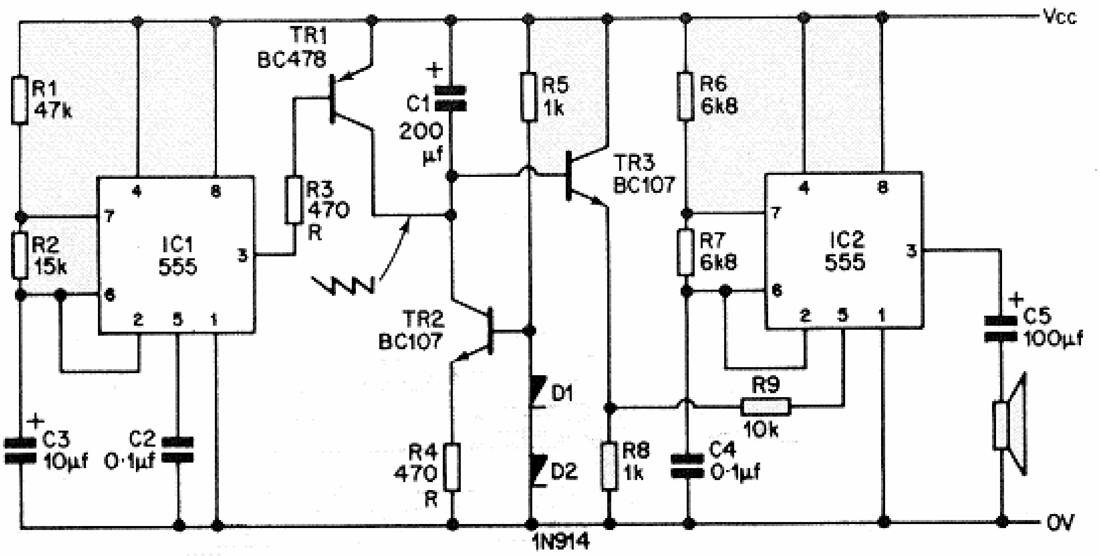
As for synchronisation this obviously cannot be tested without the PLC Simulator circuit and the PIC circuits complete. However, as said before with pre-programmed alarm bits the LCD does work correctly apart from line toggling.

Final Circuits

Shown here is the PLC Simulator circuit, with the 555 clock generator top left, the JK flip flop clock divider (divide by 16) top right and the parallel in serial out registers bottom right. The output to transmitter is shown but not the actual IC. The transmitter is simply connected as given in the datasheet in the appendices at the end of this report.



Shown here is my alarm circuit with its original values given.



Finally is the PIC receiver circuit, this is all basically designed around the PIC chip itself as per the pinout shown previously.

Parts

With design completed the next task is to generate a parts list based upon required components. Below is a categorised list of required components:

PLC Simulator

1. 2 x 8-bit PISO ICs
2. 555 Timer
3. 2 x JK Flip flop ICs (two per IC)

Transmitter

1. TWS-434 IC

Receiver

1. RWS-434 IC

PIC Receiver

1. PIC16F84A
2. 555 Timer
3. LCD Module

Other discrete components and less costly items such as LEDs are not mentioned here.

Analysis

The tests gave me the following information to work from:

1. Transmitter and receiver chips are correct for my usage
2. Simulator design is correct
3. Alarm circuit works as planned

Problems

The only problems experienced in design and testing were the following two:

1. Synchronisation of bit streams
2. Transmitter/Receiver reception

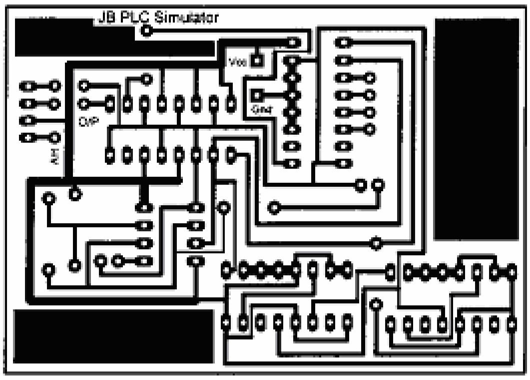
The synchronisation of bit streams is both a hardware and software issue. On the hardware side a careful calculation of clock frequencies is required and on the software side programming of “locking on” to the sequence in time to the clock is required.

The transmitter and receiver reception has one easy solution, 30cm antenna. However if that makes the design too large practically I would have to choose other ICs, for the purposes of this project though that should not prove to be a problem.

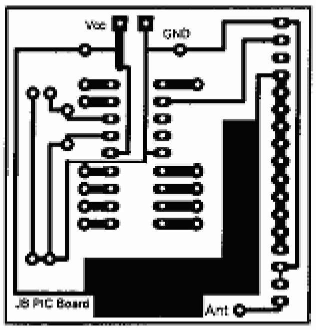
Building

PCB Design

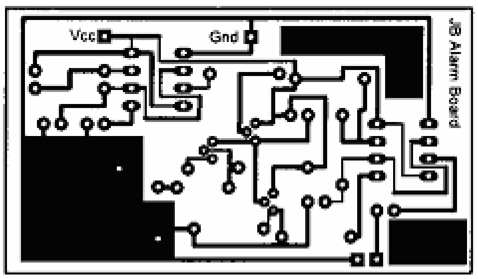
Shown here are my PCB designs, the PLC simulator (with for the purposes of this project the transmitter built in), the PIC and receiver board, and the alarm board. These circuit boards are the same as the final circuits given previously.



PLC Simulator and Transmitter

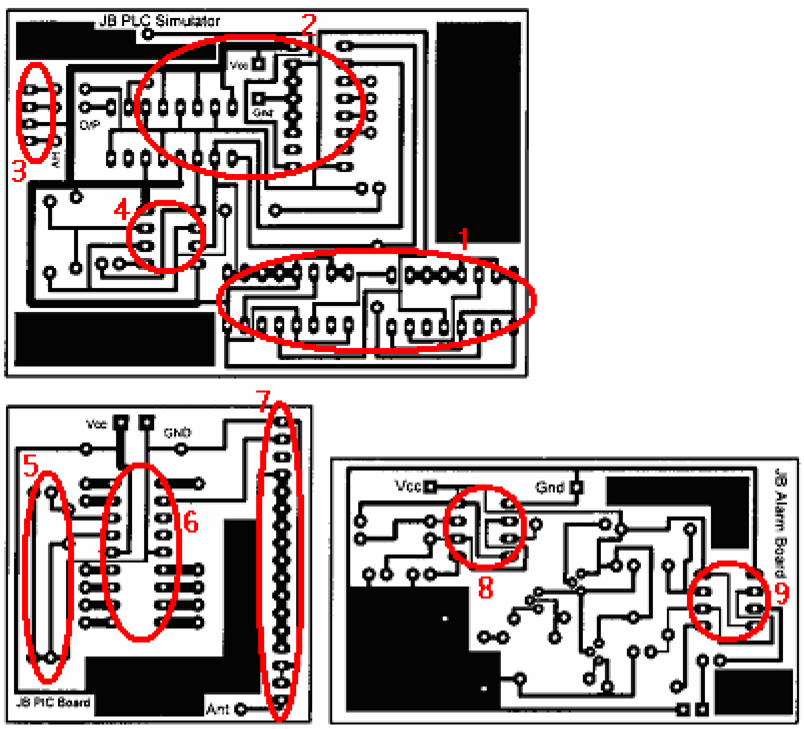


Receiver and PIC Board



Alarm Board

Shown here are the PCBs again with highlights explaining the component layout.



1. JK Flip flop chips for clock divider.
2. PISO chips for output to transmitter
3. Transmitter IC
4. 555 Clock
5. Timing components
6. PIC16F84
7. Receiver IC
8. 555 Timer
9. 555 Timer

Final Building

The only problems after designing were:

1. A track and node that had come out touching after the PCB had been created that I had to manually sever.
2. I accidentally put a 4-pin SIL in the position of my transmitter IC which is a 6-pin SIL. I did however manage to correct this by drilling two extra holes either side of the 4 pins and then using some linking wire.

Testing

First Testing

The first board I completed was the alarm board and that actually worked first time of powering up.

Neither of the other two boards worked in whole or in part.

Problems

The problems with the boards appear to be the LCD unit on the receiver and the PISO chips on the transmitter, I am however still looking into this.

Overcoming Problems

As I stated previously I have not corrected these problems yet but the problems appear component related not PCB related and so should not be too difficult to correct.

Evaluation

Working?

All of the units worked separately on breadboard and the PIC program worked mainly in the testing board.

The alarm board as stated previously does work, which being made up of mainly discrete parts I was expecting to be the least likely to work.

Keep to POA

I did keep to my Plan of Action until the beginning of April due to a rush on other work. After the end of March I started to get behind on building and testing.

Keep to Specification

My specification is correct to what I have aimed to produce with the exception of encoding and decoding. There is no encoding and decoding in my project at all. The only reason this was left out was due to cost though, explained in the following.

My project budget was £25 and I spent just under £40. This was mainly due to the cost of the transmitter and receiver ICs.

Future Improvements

If the budget of the project was higher I may have been able to use a system that would have allowed for further distance since although the datasheet for the ICs I am using says 100m this is an absolute maximum. In reality this project would use a more expensive transmit/receive pair that allow for a larger distance.

The four alarms that this system uses is quite sufficient in a lot of circumstances but perhaps six, eight or even more would be more beneficial, this is simply a matter of programming and a larger PISO setup.

The alarms that are coded into my PIC program are just that, coded into the program. As an improvement I believe the ability to adjust the alarm names after having produced the project would be a necessity. Having thought about this I decided the simplest way would be to store the alarm names in a separate EEPROM memory on the PIC of the same board and either reprogram the EEPROM from remote (computer running special software) or alter the PIC program itself to allow for a remote connection to be established to download the alarm names direct to the PIC.

One feature that could be of use is a vibrating alert so that if the audible alert needs to turned off for a meeting as an example the person using the unit could still be alerted.

A reasonably small but useful feature would be letting the PIC control the LCD backlight under different lighting conditions, or battery level, or to make it flash if there is an alarm.

Unfortunately none of these were implimented due to time and cost restrictions.

Bibliography

Internet Addresses

PIC16F84 Datasheet
www.microchip.com/download/lit/pline/picmicro/families/16f8x/30430c.pdf

RF Solutions
www.rfsolutions.co.uk

Hugh Jacks Automated PLC Information Book
http://claymore.engineer.gvsu.edu/~jackh/books/plcs/

Books

Electronics Communication Techniques (Fourth Edition)
Paul H. Young
ISBN 0-13-779984-5

555 Projects
E Parr
ISBN 0 85934 047 3

Appendices

PIC Program

Program Flowchart

PROJECT.ASM

Datasheets

Microchip - PIC16F84

RF Solutions Encoder/Decoder - HT12A, HT12E

Rentron Tx/Rx - TWS/RWS434

RF Solutions Rx/Tx - RTF/RRFQ2

Maplin User Encode/Decode - N49AU/N50AU

RF Solutions Encoder/Decoder - RF600E/D

Phillips PISO - 74HC/HCT165

Powertip - PC1602 LCD Module

PLC Data

Serial Comms