LDPS IV Lab Report

FM Radio Receiver

Vilma Wilke
Le Huy Hoang
Umair Hassan
Joshua Tyler

Table of Contents

1 Project specification and objectives 2
2 Project planning 3
  2.1 Writing code 3
  2.2 Debugging code 3
  2.3 Hardware design 4
    2.3.1 Introduction 4
    2.3.2 FM receiver IC (AR1010) 4
    2.3.3 Voltage regulator IC (LE33CZ) 5
    2.3.4 PIC microcontroller (PIC18F6490) 5
    2.3.5 Audio amplifier IC (TDA2822) 5
  2.4 Hardware construction 6
3 Preliminary design 7
  3.1 Initial hardware design 7
    3.1.1 Audio amplifier output 7
    3.1.2 Voltage regulator 7
    3.1.3 The I^2C connections 8
  3.2 Initial code design 8
4 Final design 10
  4.1 Final circuit diagram 10
    4.1.1 Audio amplifier output and voltage regulator 10
    4.1.2 The matrix board design and I^2C connections 10
    4.1.3 The LCD connections 11
  4.2 Final code design 12
5 Documentation 13
  5.1 Test at specification 13
  5.2 Operating manual 13
6 Novelties and selling points 15
7 Technical challenges 16
  7.1 Layout of the hardware 16
  7.2 Wiring on the matrix board 16
  7.3 Debugging the I^2C interface 16
8 Conclusion 17
9 Appendix 18

1. Project Specification and Objectives

The objective of this project is to build a prototype FM receiver which can tune over the UK broadcast band by using a PIC microcontroller to operate an FM receiver chip. The device should also make use of an LCD screen to provide information to the user.

The provided specification of this project states that the radio must be:

1. Tuneable approximately over the UK broadcast band (87.5 - 108 MHz).
2. Able to give stereo (L & R channels) audio outputs.
3. Capable of driving a normal set of 35 ohm impedance headphones.
4. Powered by a (reasonably sized) battery.
5. Portable (including an antenna).
6. Constructed using components and facilities that are either in the labs or are readily obtainable at reasonable expense.
7. Able to display the current frequency on an LCD display.

2. Project Planning

A total of 5 week labs (6 hours each) were allocated to us to finish this project. It means we had 30 hours of laboratory time to tackle the problems. The lab was also available to us for a further 3hours per week for 4 further weeks, had we required additional time to complete the project. The project is an exercise of both software and hardware design. Our group, which comprises of 4 people, divided into 2 sub groups to work on hardware and software at the same time. The hardware group had to complete the Audio amplifier, regulator, LCD display hardware connection. The software group focused on PIC programming using MPLAB.

2.1 Writing Code

The code was a major part of this project and so planning our approach to it was critical to ensure success.

We approached this task in a number of stages. Firstly we spent a week familiarising ourselves with the language and development environment, this allowed us to get to grips with how the C language was used to control the PIC. We also looked over the skeletal code which was provided to interface with the AR1010 and familiarised ourselves with that. We planned to do all of this on the first week because during that time the specific PIC which we would be using for the project was not yet available, so we had to use a development board.

Following this initial familiarisation we started development of the actual project. Josh and Vilma worked on separate features in parallel and then put our work together at the end of each session. This approach worked because we both used functions to write modular code. We started off working on the most critical functions, such as functions to read the pushbutton input and communicate with the AR1010. Once all of this basic functionality had been completed, we moved on to writing more advanced things such as the code for the LCD display.

At each stage we made sure to include test code which could ensure each functional unit worked without having to depend on other parts of the project.

Where necessary we liaised with the hardware team to ensure the hardware was designed and constructed in a manner which would make the software easy to write. An example of this is when we decided how the LCD would be wired up, as will be explained later, in addition to the fact that we requested the AR1010 and buttons be wired up first, so that we could begin to debug the project as soon as possible.

2.2 Debugging Code

One major difficulty in the FM radio receiver project was the merging of hardware and software functionality. In order for the final product to function as intended, a systematic way to search and find the source of error had to be adopted. This would ensure that testing could be done efficiently and with little guesswork.

Our group effectively split into two teams, with one focusing on hardware and the other one on software. The first line of error checking the software came in the form of having two members. By cross-checking the code and logic with each other, obvious issues could be caught in a very efficient manner before the program was even compiled.

However, no amount of looking at the code would be enough to fully test for errors. However, the code was often finished faster than the hardware was able to accommodate it, and waiting for the hardware to catch up would cost valuable time. Thus, simulating the hardware and observing the logic was the best option. The program and compiler used, MPLAB, had the capability to simulate the operation line-by-line. This proved to be invaluable in debugging, as it was possible to observe the logic in real-time.

Even with that, it was impossible to test the functionality of the LCD display, as there was no way to simulate such a thing. Thus, it was inevitable that once the hardware was fully functional and integrated with the software, almost all bugs were display errors. But as a result of our previous efforts, the errors that remained were minor and could be quite easily resolved. Final debugging in this part was mostly done with one person programming, with the other one going through every conceivable procedure to check when and where errors and bugs would occur. This proved to be more efficient than two people attempting to debug the code simultaneously.

By the end of the deadline, the procedures we took appeared to have had a clear result. The final integration of hardware and software resulted in a product that works to, and above specifications. The time saved in debugging allowed for the opportunity to test, debug, and implement some extra functionality, such as a scrolling text function and utilising the seek capabilities of the AR1010 chip. Furthermore, all the precautions taken while debugging resulted in a stable code.

2.3 Hardware Design

2.3.1 Introduction

Before jumping in and constructing the hardware, it is first important to research and understand what each of the components do and how they work. We therefore conducted research into the various components before the first session and here are the results.

The FM receiver is the electronic device that receives the radio waves and converts the information carried by them to FM waves. It is used with an antenna. In the FM project there are two main chips to control the receiver. They are FM-receiver chip and PIC Microcontroller. The hardware includes the audio amplifier, the connection of AR1010 receiver to 3.3 volt regulator, the connection of AR1010 receiver to PIC18L6490 and the connection of PIC18L6490 to LCD display screen and button panel. Audio amplifier TDA 2822 is used to amplify output sound signal from the FM receiver chip in order to drive the 35Ω headphone.

2.3.2 FM Receiver IC (AR1010)

It is a single chip stereo FM Radio receiver. The AR1010 chip is received from AIROHA. It's a package of 24pin and is only 4mm^2. AR1010 can be used to encourage a high FM bands from 76 to 108MHz. It merges Low-noise amplifier (LNA), mixer, oscillator and low drop out regulator (LDO). The AR1010 requires antenna to receive the FM radio signal. The chip is powered by 3.3 V DC supply and VHF FM module into signal. It is controlled by microcontroller PIC18LF6490.

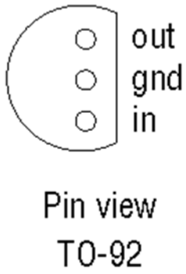
It is used to receive FM signals and demodulate them. It then outputs them to an audio amplifier. Besides that it allows setting the volume level suitably. It tunes to any specific frequency. However, for the special purpose, the AR1010 board has 10 pins. The function of each pin is shown below:

PIN NO.	FUNCTION
PIN 1	ANTENNA
PIN 2	Not connected
PIN3	LV_{OUT}
PIN4	RV_{OUT}
PIN5	V_{SS}
PIN6	V_{CC}
PIN7	BUSEN
PIN8	BUSMOD
PIN9	SCL
PIN10	SDA

Features:

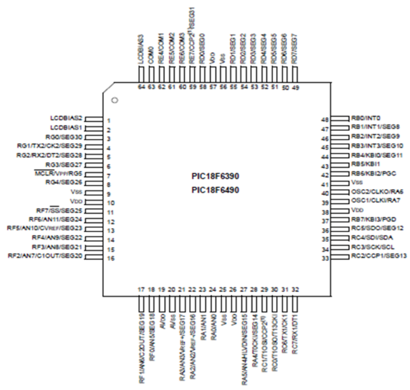
- FM band support 76 to 108MHz.
- Highest integration level with minimized external BOM cost.
- Synthesizes the frequency with the integrated voltage controlled oscillator (VCO) and automatic frequency control (AFC).
- Integrated XO with external reference clock input or external 32.768 KHz crystal.
- Can be used to measure the signal strength.
- Programmable de-emphasis time constant (50/75us).
- Adaptive noise suppression.
- Analogue output with volume control and line-level outputs.
- Serial control interface for 2-wire and 3-wire modes.
- Embedded seek tuning function.

2.3.3 Voltage Regulator IC (LE33CZ)



The voltage fed into the AR1010 should be 3.3 volts, but the supply voltage is 4.5 volts from three AA batteries, so we use the LE33CZ regulator to reduce the voltage to 3.3V. The LE00 regulator series are very low drop regulators. The very low drop voltage (0.2V) and the very low quiescent current make them particularly suitable for low noise low power applications and especially in battery powered systems. The package of LE33CZ we used is TO-92 which has 3 pins.

2.3.4 PIC microcontroller (PIC18F6490)

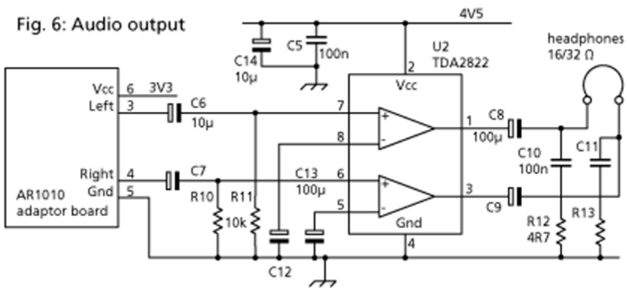


The microcontroller chosen, a PIC18F6490, is widely available, can be programmed using an in circuit serial programmer, has 64 pins and an operating voltage range of 2 to 5.5V. The function of it is that it is a processing core, and it has memory and programmable input or output abilities. Furthermore, A LCD driver module is integrated in to the microcontroller which can drive LCD panel directly with 48 segments.

PIC18F6490 features an instruction memory of 16KB, 768 Bytes of RAM, CPU speed of 10 MIPS, Internal oscillator with frequency range from 8 MHz to 31 MHz and it also supports digital Communication Peripherals such as I2C and USART.

2.3.5 Audio Amplifier IC (TDA2822)

In order to drive a normal 35ohm impedance headphone, an audio amplifier, TDA2822, is chosen. It is a small power amplifier that can drive two channels. A 3v energy supply can offer 20mW in 32Ohms per channel to generate headphones.



TDA2822 audio amplifier amplifies the audio output signal from the AR1010 chip. Connection of audio amplifier TDA2822 to AR1010 chip is shown below. C1 and C2 removes DC bias coming from receiver chip and C6 and C7 act as coupling capacitors. The AR1010 is capable adjusting the volume so the addition of a volume potentiometer was unnecessary.

2.4 Hardware Construction

For the hardware of the FM receiver, lots of work needed to be planned and carried out. This work consisted of items such as circuit's block design, connection design, matrix board layout as well as simply soldering components. Therefore, an effective task plan was created and followed to make sure everything stayed on track:

During the first section of the project, after getting all necessary components for the project, the amplifier output circuit was built on the breadboard, then tested to ensure it gave a clear signal at the output. After that, the block of the voltage regulator was also built to connect to the audio amplifier on the same breadboard.

During the second section of the project, a circuit design on veroboard of the voltage regulator integrated to the audio amplifier was produced after checking the operation of the circuit on breadboard. The reason veroboard was picked for this over matrix board or PCB to build these circuits on was because these circuits were not too complicated and could be drawn manually. Veroboard was also readily available in the main laboratory, making it easier and quicker to build as well as to check the operation of each block after finishing.

During the third section of the project, the connections between the audio amplifier, voltage regulator and the AR1010 chip were soldered and checked carefully before starting to connect the I^2C connections which was located on the matrix board between the main processor chip PIC18F6390 and the given AR1010 board.

During the fourth section of the project, the I^2C connections were debugged. After that, the push button connections were set up and soldered on the matrix board.

During the fifth section of the project, the table mapping the pins of the PIC and the LCD connections was produced and these wiring connections were soldered by using the wiring pen. That made the process of soldering slower than was expected. Moreover, after finishing soldering the LCD did not display segments correctly and we, therefore, had to re-design the pin connections between these two components and redo the soldering.

During the last section of the project, we finished re-soldering the LCD to PIC connections and checked all the minor problems relating to hardware before finalising the code and testing the receiver to ensure that it met specification.

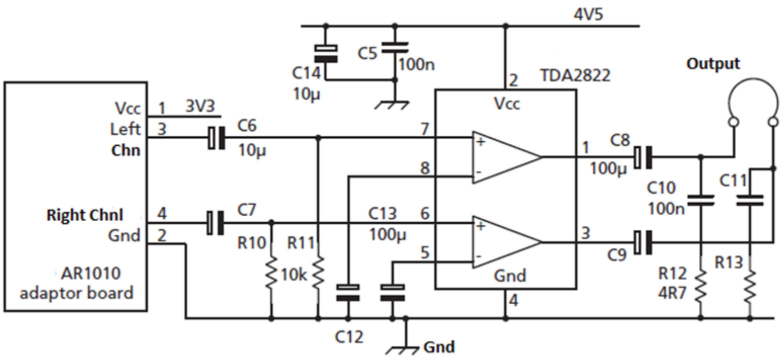
3. Preliminary Design

3.1 Initial Hardware Design

With respect to the hardware part, the FM Receiver is a combination between quite a few circuit blocks such as audio output, voltage regulator, AR1010 adapter and the LCD connections block.

3.1.1 Audio Amplifier Output

Initially, the integrated audio amplifier chip TDA2822 which is designed and commonly used for portable cassette players and radios is provided. Therefore, our task is to build the audio amplifier circuit based on that chip as the following circuit design:



It is quite a simple circuit and is loosely split into input from the AR1010, process and output stages as you look from the left hand side of the circuit across to the right.

On the top of the diagram, we have the batteries which supply the power to the circuit. We chose to use three 1.5V AA batteries (3x1.5=4.5V) from an early stage, because the batteries and holders are cheap and readily available in the laboratory.

On the far left, the input signals fed by the AR1010 receiver go from the right and left channels. The given AR1010 adapter board cannot handle the voltage of 4.5V from the main power because it will heat up and possibly break if that high a voltage is applied. Therefore, in order to safely operate, a voltage regulator which can step down the voltage magnitude from 4.5V to 3.3V to feed into the AR1010 is required. This block design and description will be mentioned later.

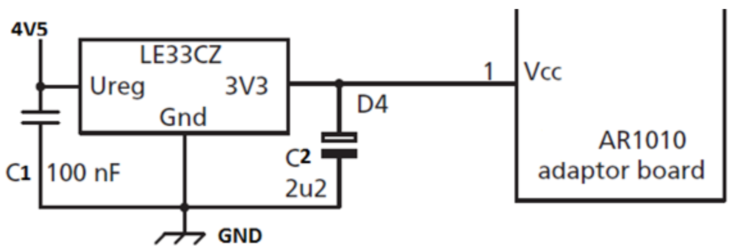
The right and left channels from the AR1010 adapter board will produce DC bias. Therefore, it has to be removed by using the coupling capacitors C6 and C7 connected to the input pins 6 and 7 of the TDA2822, respectively. These two inputs are pulled down to 0V through two pull-down resistors of 10k Ohms.

In the centre of the diagram is the integrated audio amplifier chip TDA2822. It produces two output channels which can be directly coupled to the speaker through decoupling capacitors C8 and C9.

On the right hand side of the diagram we have the output stage which is supposed to provide a sound signal through a normal set of headphone which has resistance in range between 16-32 Ohms.

3.1.2 Voltage Regulator

As mentioned above, the AR1010 cannot handle the source voltage of 4.5 volts so it is necessary to have a voltage regulator connected to the main source in order to make sure that after going through the voltage regulator, the output voltage recorded has to be less than 3.5 volts. And the circuit diagram for the voltage regulator which was provided is below:



As can be seen on this diagram, the circuit for the voltage regulator is quite simple. We just need to apply 4.5V to one pin of the regulator, and a second pin outputs 3.3V. This regulator is particularly used for low noise, low power and battery-powered systems. A capacitor of 2.2 micro Farads is used in the output stage of the voltage regulator to make sure the output voltage is stable. The output voltage to be provided for the AR1010 board will be 3.3 volts, as required.

3.1.3 The I²C connections

After building two blocks of the audio amplifier and voltage regulator, in the next step, the I²C connections will be connected between these two blocks and the main PIC18LF6490 by using a five-way connector. This task is carried out with the help of the given diagram as below:

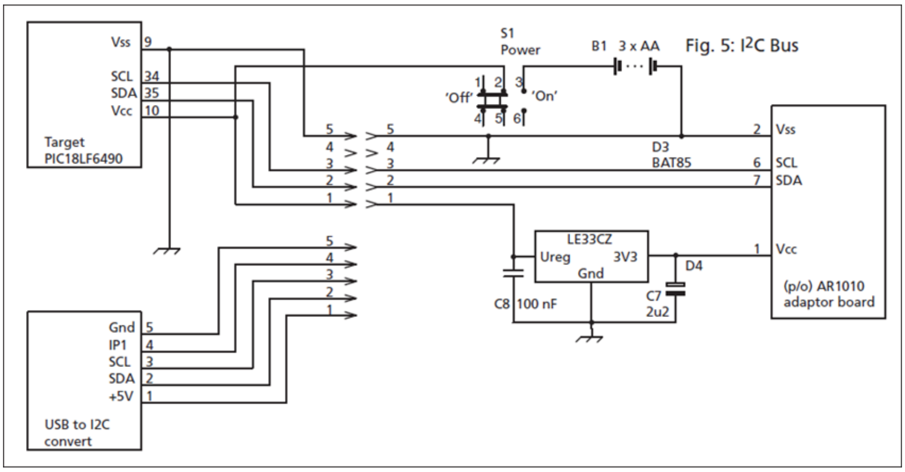


Fig. 5: I²C Bus

In the above diagram, the main PIC will be powered at 4.5 volts and this is connected to the AR1010 which is run at 3.3 volts by using connections 1 to 5 in the arrangement above. One thing is noted here is the physical I²C bus which is represented as two wires called SCL and SDA. SCL is a clock line and can be used to synchronise all data transferred over the I²C bus. SDA, in the other hand, is the data line which is responsible for carrying all data transferred over the I²C bus. These two lines are wired to pins 6 and 7 on the AR1010 adaptor board, respectively.

3.2 Initial code design

To ensure a final product that worked as intended, it was important to ensure that we had good, stable code to operate the AR1010 chip based on user input. In order to accomplish this, good time and resource management would have to be employed. The programming also had to be done in an environment that was already somewhat familiar and would not require extensive amounts of relearning.

To summarise the parameters we had to comply with, a summary of what we needed to accomplish was necessary for good organisation. In order for the FM radio receiver to function to specification, a microcontroller is needed to perform several important functions. The PIC18LF6490 supplied was to be programmed using the C language to be able to interpret input in the form of button presses. This data was then to be sent to the AR1010 FM receiver chip, and finally displayed to the user on an LCD screen. Depending on the button press, the microcontroller should send the appropriate commands to the LCD and AR1010 chip. To meet minimum specifications, the final code should direct the microcontroller to adjust volume, frequency, and switch between a series of present channels.

We were provided with a skeleton outline of the final code to give us an idea of what we needed to do. This skeletal code lacked functionality in all subroutines, but would set up a connection to the AR1010 chip to begin communications. In order for the final product to function properly, subroutines for handling button press and tuning the LCD were needed, as well as several other routines to handle manual tuning, pre-set tuning, and volume control.

From the skeleton code, we could establish some basic parameters that would influence how the workload was to be approached. Several functions were in a similar vein, working to adjust a parameter once in a direction, then return the result. These functions included the volume control, frequency change, and channel change. These were grouped as a single task, as the code was repetitive and would be easier to debug if a similar code was used across all of these functions.

The rest of the functions were more complicated and lower level than the button response subroutines. These included checking for button press and displaying to LCD. There was some experience in these areas from a previous project, and that was something our group could draw upon to properly approach these difficult tasks.

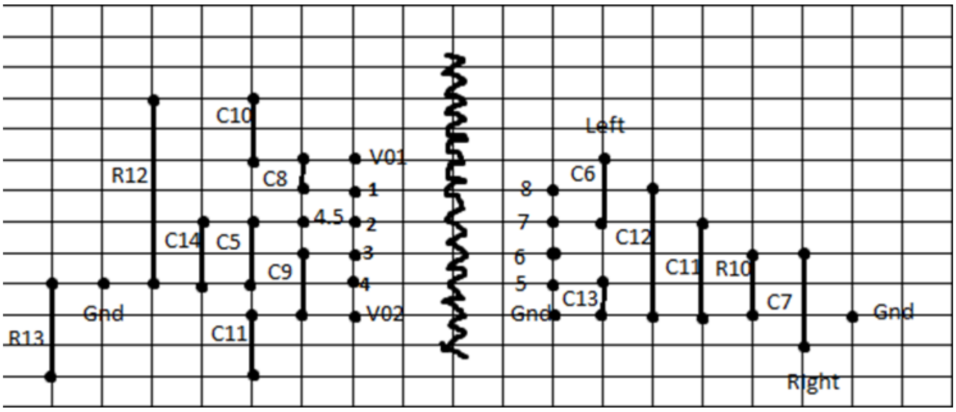
As our group had two people as dedicated programmers, it was important to ensure that there was good communication and methods of merging the code. Having each member work within subroutines made transferring code much easier, as there was no specific coding that needed to be carefully merged. Rather, the new subroutines were copied into a master project, where the routines could be called within the main function as they were needed. Our design was such that the custom functions did not call each other, which minimised time spent locating minor syntax errors and bugs.

4 Final Design

4.1 Final Circuit Design

4.1.1 Audio amplifier output and voltage regulator

Based on the initial circuit design of the audio amplifier, we built and carefully checked its operation. After that, we decided to build the audio amplifier on the veroboard with the following layout:

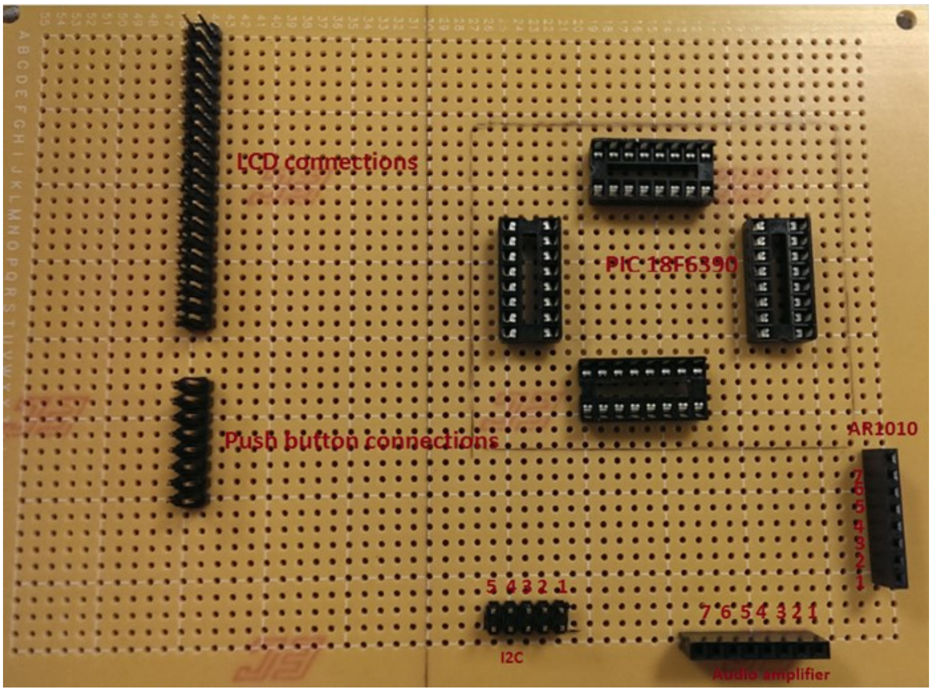


The TDA2822 chip is placed in the centre of the board and it is surrounded by the other components such as capacitors, resistors, and connecting wires as noted in the circuit diagram The size of the veroboard is quite small so it can be easily placed next to the matrix board which contains the main chip and other connections for the operation of the FM receiver.

With the fact that the circuit for the voltage regulator is very simple, just only three components: the voltage regulator and two capacitors, were also placed on the veroboard in the top right corner.

4.1.2 The matrix board design and I²C connections

After finishing the design for two blocks: audio amplifier and voltage regulator on veroboard, the next step is to design the arrangement on the matrix board which almost all components are placed on. The final design layout of the matrix board in shown in the below picture:



For the AR1010 adapter: all pin connections are placed on the matrix board as the picture above where pin 1 to pin 7 are placed from the bottom to the top of the holder. Then then connect to the Audio amplifier which is placed at the bottom corner of the matrix board. The I²C connection holder is responsible for all connections between the PIC 18F6390 to the Audio amplifier as well as the

AR1010. All these connections through this I²C header are noted in the table below:

I²Cpin	Connection from	Connection to	Note
Pin 1	Audio amplifier : Pin 5	PIC : Pin 10	4.5 Volts
Pin 2	AR1010: Pin 7	PIC: Pin 35	SDA line
Pin 3	AR1010: Pin 6	PIC: Pin 34	SCL line
Pin 5	AR1010: Pin 2	PIC: Pin 9	GND

Followed by the I²C header connections, the connections between the AR1010 and audio amplifier are also listed in the following table:

AR1010 Pin	Audio Amplifier Pin	Note
Pin 1	Pin 1	3.3 Volts
Pin 2	Pin 2	GND
Pin 3	Pin 3	Left channel
Pin 4	Pin 4	Right Channel

4.1.3 The LCD Connections

The pin connections between the PIC and the LCD display is listed in the following table in which each pin connection represents a segment on the LCD display:

PIC Name	PIC Pin	LCD Name	LCD Pin
RD0_SEG0	58	3A	35
RD1_SEG1	55	3B	37
RD2_SEG2	54	3C	39
RD3_SEG3	53	3D	40
RD4_SEG4	52	3E	38
RD5_SEG5	51	3F	33
RD6_SEG6	50	3G	31
RD7_SEG7	49	DP1	21
RB1_SEG8	47	2A	27
RB2_SEG9	46	2B	29
RB3_SEG10	45	2C	34
RB4_SEG11	44	2D	32
RC5_SEG12	36	2E	30
RC2_SEG13	33	2F	25
RA4_SEG14	28	2G	18
RA5_SEG15	27	DP2	28
RA2_SEG16	22	1A	17
RA3_SEG17	21	1B	19
RF0_SEG18	18	1C	26
RF1_SEG19	17	1D	24
RF2_SEG20	16	1E	22
RF3_SEG21	15	1F	15
RF4_SEG22	14	1G	13
RF5_SEG23	13	DP3	36
RF6_SEG24	12	K	10
RF7_SEG25	11	COL	20
RG4_SEG26	8	Z	01
COM 0	63	X	02
COM 0	63	Y	08
COM 0	63	COM	04
COM 0	63	COM	06

This scheme was chosen so that the pins representing one single digit and accompanying decimal point are mapped to a single LCD_DATA register in the PIC. This means that code to write a letter or digit can be reused for each of the three digits. This greatly simplifies the design of the code.

4.2 Final Code Design

The final code remains close to the original design of the code, but with a few key differences. These for the most part are focused on technicalities in communicating with the AR1010 chip. There are also a few subroutines added at a later stage to accommodate extra functionality that was not mandatory, and thus not originally planned out.

The two most notable differences in the original design are the addition of two extra features. These are the seek function and the scrolling text function. There are also several changes done in order to accommodate proper functionality, something that is especially noticeable in the volume control function. However, the general structure was adhered to closely and only require some minor fine-tuning.

One of the last major changes to the code before extra additions was to the volume control function. The volume control required that two bits in the regImg[adr] array were altered to a corresponding pair of values. Not only did these values have to be converted from the values supplied in the AR1010 user manual to ones that the AR1010 would function with when received, but the values only consisted of two different parts of two separate strings. In order to only write the correct part of regImg[], the rest of the selected value had to be masked off to ensure that they were not mistakenly altered.

Several minor changes were also made to the structure of the skeletal code that had been provided. Tuning and volume control subroutines altered to return a variable held in the main routine as a local, since making them global was unnecessary. The functions were rewritten to have two inputs (current value and direction of change) and return the new value to store in a local variable in the main function.

Seek functionality is a property already included in the AR1010 FM receiver chip, and only required proper software implementation on our part to have a positive result. We used a similar method as with volume control to write to the bits needed and masking off the parts of the string we did not need to alter. Our previous experience with this in the volume control function made this task ultimately simpler to approach.

As a final aesthetic addition to the FM radio receiver's functionality, it was decided that a function for scrolling text could be included. This was decided towards the end of the project, when all else had been finalised and debugged. The scrolling text function uses the limited display to show the letters of the alphabet. The limitations being the display not being ideal for such display, as well as insufficient hardware to display anything besides the uppercase writing. When the function is called, it displays a string by scrolling the text across the LCD display to work around the small display. The text data is obtained character-by-character from a data table dictating which segments to use to represent a character. Whilst the function itself does not handle stings with a whitespace, the "[" character has been written into the library as a whitespace. In the final code, the scrolling text function is called in two instances: when starting up the device ("HELLO" is displayed) and when switching through the preset channel list (In which case, the name of the channel is displayed before returning the display to showing the frequency).

5 Documentation

5.1 Test at Specification

In order to ensure that our FM receiver met the necessary criteria, at the end of the last session we tested our FM receiver against the original specification. The results of this testing are shown below

Specification point no.	Specification point	Was the point met?
1	Tuneable approximately over the UK broadcast band (87.5 - 108 MHz).	**Yes**, when tested the receiver can be tuned to frequencies over the entire range.
2	Able to give stereo (L & R channels) audio outputs.	**Yes**, the AR1010 has a stereo output, which coupled with a stereo amplifier means that the project can output a stereo signal.
3	Capable of driving a normal set of 35 ohm impedance headphones.	**Yes**, the audio amplifier is capable of driving 35 ohm headphones at a good volume.
4	Powered by a (reasonably sized) battery.	**Yes**, the entire project is powered by three AA batteries.
5	Portable (including an antenna).	**Yes**, the chassis is compact and all components are securely fastened so the radio is portable. The antenna connects to a standard co-axial connector, so a portable antenna could be taken with the radio.
6	Constructed using components and facilities that are either in the labs or are readily obtainable at reasonable expense.	**Yes**, all components used were already stocked in the lab. No external facilities were used either.
7	Able to display the current frequency on an LCD display.	**Yes**, the display shows the frequency, amongst other information.

As shown in the table, the FM receiver meets all the specification points. In addition to this it also implements extra features.

5.2 Operating Manual

This FM radio receiver comes with several basic applications that are expected of a radio to function, but has some extra functionality in addition to this. Included is the ability for the FM radio receiver to switch between a series of preset channels (though manual tuning is also possible) within a frequency range of 87.5 – 108.6MHz. Also available is the option to seek for the nearest channel to the current frequency, with the user deciding whether to look with increasing or decreasing values.

The FM radio receiver is operated exclusively through the eight buttons located on its front. The buttons correspond to the following functions:

1. Volume up
2. Volume down
3. Manual tune up
4. Manual tune down
5. Preset list increment
6. Preset list decrement
7. Seek up

8. Seek down

Included is a list of six preset channels for the user to switch between. The user's position in the list is saved shall the user wish to use any of the manual frequency shifts. Using the command buttons to change the channel again in such circumstances will switch the frequency to the next/previous in the list. The preset channel frequencies are as follows:

1. **87.5** (Unknown one)
2. **88.1** (BBC two)
3. **90.3** (BBC three)
4. **96.4** (EAGLE)
5. **104.6** (BBC one)
6. **112.9** (Unknown two)

Changing any of the values will result in the LCD display showing the new value in question for a few moments before reverting back to the default setting of showing the current frequency. The volume operated on a scale of 1 to 18, with 18 being the loudest volume. When switching between preset channels, the name of the channel will be displayed on the screen. The text displayed for each channel is written in the brackets in the frequency guide above. This text will not display for channels found via manual tuning.

6 Novelties and Selling Points

Once we had finished development of the main functionality of the radio, we added several additional features. These features were:

- Scan functionality so that the user can find the next radio station without having to know the frequency. This feature works by creating a function in the PIC code with interfaces with the AR1010 telling it to use the scan feature. The code for this was non-trivial as quite a few registers needed to modified set in the AR1010 to get it to work.
- A preset station feature. This feature works by hard-coding the frequencies of several local stations into the PIC, then when the user selects to go the next preset the pic sends the AR1010 the frequency of the next station on the preset list. We wanted to have a user settable preset list, but unfortunately this was not possible due to the PIC not having EEPROM memory on board.
- A volume readout feature. When the user changes the volume, the PIC displays the volume on-screen.
- A scrolling textual station readout feature. When the user changes to the next preset station the name of that station is displayed on the screen as scrolling text. This feature was made possible by hardcoding the necessary register configurations to display each character on a seven segment display into an array, then looking it up on the fly when a character needed to be displayed.
- A Perspex mount for the circuitry. We improved upon the provided chassis by mounting our circuit boards on a Perspex base. This avoids potential issues that may arise from insecure circuit boards.

7 Technical Challenges

The main technical challenges in this project were:

7.1 Layout of the Hardware

We had to make a very complicated circuit with more than 64 connections just from the PIC. We had to make a design on a small matrix board so that it can come with in the box of the FM receiver. It was really hard to make a space for AR1010 in the board while keeping the circuit less complicated. We made another small circuit for AR1010 on a very small veroboard and connect it with the matrix board.

7.2 Wiring on the matrix board

We worked on a matrix board because the PIC was not designed to be work on the vero board. We used road runner soldering for the connections as we had to made 64 connections from the PIC and the size of the board was small. We numbered the back of the board with stickers so that we can see the pin number before soldering. As the pins were very close to each other soldering was also a challenging job. The problem with the soldering was that once it is done it is really hard to track it because more one wire is in the same path. We had to rebuild connections several times because we made minor mistake with the numbering and in checking the connections.

7.3 Debugging the I^2C interface.

The I^2C interface between the PIC and AR1010 was key to the entire project, however it is something we initially had problems with. We solved these by taking a logical approach. We first stepped through the code with the PICKIT debugger to determine whether the problem was a software or a hardware one. From this we discovered it was a hardware problem. Following this we then traced the I^2C power communication lines and performed continuity tests to find that the problem lay with a faulty solder joint.

8 Conclusion

In conclusion the FM receiver performs to specification, as well as adding other features to exceed specification. In this respect, the project has been a success.

In addition, we have all learned more about hardware and software design from this project, and especially the interaction between hardware and software, the problems that can occur due to it and the ways in which those problems can be overcome.

9 Appendix

The appendix for this project consists of the C code, and header file for this project (main.c and fm.h respectively). They compile under MPLAB X to make the instruction code for the PIC.