1. FM Receiver Project

Lab Location: Main UG Labs
Teams: 3-4 students per team

Most electronic devices now depend on a mixture of hardware and software. It is therefore essential to learn both software and basic electronics skills. The FM receiver project aims to introduce you to combining software and hardware design. You will be provided with an FM receiver chip, which needs to be operated using a microcontroller. Your goal is to produce a prototype FM receiver, making use of a PIC, an LCD display, and an audio amplifier chip, as shown in Figure 1-1. For the basic function of the FM receiver, you need to use the components provided from the lab. Please note that you may be given a slightly different component than what shown in the figure. This section of the manual provides an introduction to the production. Further details will be posted on SurreyLearn.

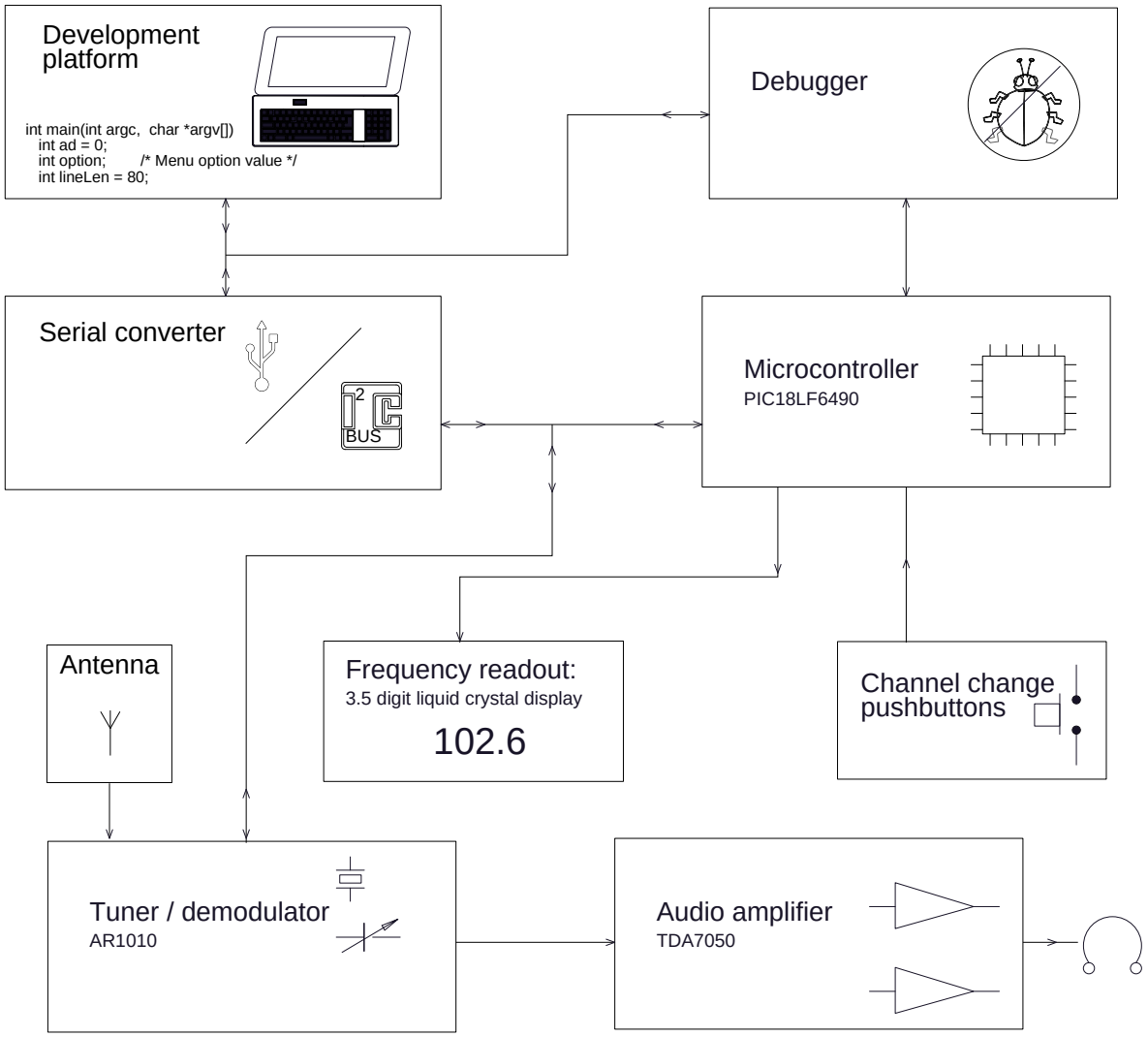


Figure 1-1: Basic FM Receiver Schematic.

1.1. Getting Started

The project should be carried out in groups of 3-4. It is recommended that you spend the first session planning and allocating tasks for each person, as you did for the project in the first

semester. One way to approach the project is to break it down into three phases: research, design, and prototyping/testing.

Research Phase

To complete the project you will need to pick up new knowledge and skills. This requires you to perform some research on both circuit design, and on PIC programming. The SurreyLearn page will contain example circuit diagrams, plus details about the FM chip and other components that you will be given. This is a good place to start when considering how to approach the circuit design.

If you will be responsible for programming, then your first task will be to learn how to write basic C programmes for a PIC. To do this, you will have to first learn how to use the Integrated Development Environment (IDE) called MPLAB, produced by a company called Microchip. Please see the SurreyLearn webpage for some links on getting started with MPLAB. Once you have familiarised yourself with the MPLAB software and some basics of PIC programming, you should proceed to study the example C code for the FM chip (i.e. the skeleton code). Additionally, you are provided full source code of "pc2fm" which is used to control FM chip using a PC. In the source code, you can see how I2C is used to control FM chip. You can follow the same logic for your PIC programming. Note that you will also have to write code to achieve the following:

- Output digits to the LCD display
- Detect and take action in response to button pushes (e.g. on/off, tune up/down)

Design Phase

Once you understand how to programme a PIC and the circuit theory, you can start to design your software and your circuit. Although it is a good idea to split the work into hardware design and software design, you will have to be careful to communicate with each other, so that your software is capable of working with your hardware and vice versa. Your team should come together regularly to review progress. Try to think about how you might troubleshoot during the design process, so that your design does not make it difficult to track down problems. Also, make sure to keep a note of your planning and reasons for design decisions, as these will be examined at the end of the project.

Prototyping and Testing Phase

An Electronic Prototype System (EPS) will be made available to you for prototyping your circuit. Please see the SurreyLearn page for more details about this.

1.2. Expectations and Marking

At the end of the project, you will be expected to:

- Demonstrate a working prototype FM receiver
- Present a technical report, which describes the design of the circuit, and explains the reasons and theory behind the design

If you are not fully successful in producing a working prototype, you will be judged on the milestones you have achieved during the lab session, design quality, and what you have learnt while working on the project presented in your report. ***A non-functioning FM receiver does not necessarily mean that you will fail.*** It is important that you keep your progress record updated.

1.3. Taking It Home

Unfortunately, we are not able to let you take all parts of the FM receiver home. This is not down to cost, but rather the time taken to prepare some of the components for your use (e.g. mounting of FM chips). We simply do not have the resources to replicate the considerable effort put into this preparation for each new year group. If you wish to take some parts home, then please enquire in the lab office to find out what you can take.

1.4. Changing the Design

Ideally, you should use the given components to build the FM receiver. However, you may enhance your design by replacing the allocated components with more advanced components. If you wish to replace any existing component, you must seek approval. You can find the Component Replacement Form in the handbook.

In the Component Replacement Form, you need to provide a valid reason for the replacement. A replacement should lead to more challenging design. Any proposal that simplifies the design and reduces challenges (for example reduced wiring, simplified programming, etc.) will not be approved. We may also reject your proposal if we do not believe that your proposal is practical (for example, requesting some significant change in the middle of the semester, proposing multiple microcontrollers in a single design for no good reason, etc.).

1.5. Additional Materials

Additional materials (datasheets, instructions from component manufacturers, video guidance, etc.) are given in Surreylearn.

USB-I2C USB to I2C Communications Module

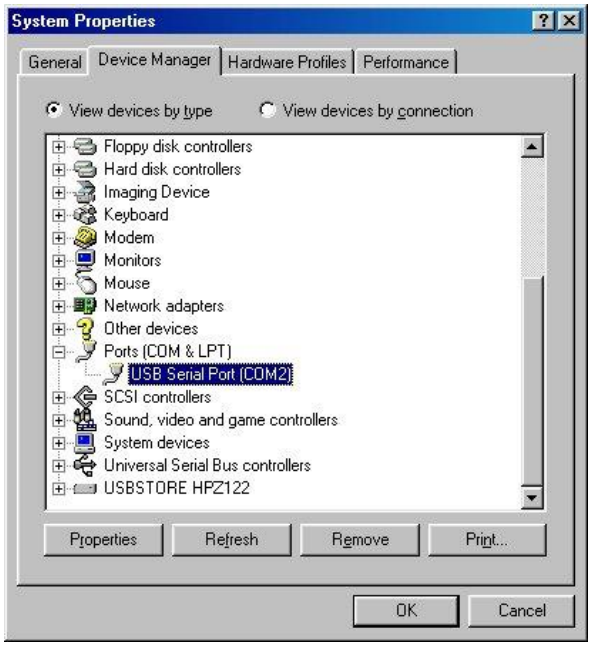
Technical Specification

The USB-I2C module provides a complete interface between your PC and the I2C bus. The module is self powered from the USB cable and can supply up to 70mA at 5v for external circuitry from a standard 100mA USB port. The module is an I2C master only, not a slave.

First Step – Test the Drivers

The USB-I2C module uses the FTDI FT232R USB chip to handle all the USB protocols. The documentation provided by FTDI is very complete, and is not duplicated here. Before using the USB-I2C, you will need to install FTDI's Virtual COM Port (VCP) Drivers. These drivers appear to the system as an extra Com Port (in addition to any existing hardware Com Ports). Application software accesses the USB device in the same way as it would access a standard Windows Com Port using the Windows VCOMM API calls or by using a Com Port Library. Drivers are available for Windows, Apple, Linux and Open BSD systems directly from the FTDI website (http://www.ftdichip.com/).

We have installed the drivers on the computers in the laboratory. If you use your own computer, you should get and install the drivers now from FTDI website, before you connect the USB-I2C to your computer. Note that newer Windows OS will automatically detect and install the drivers once you have plugged in the USB-I2C module to your computer.



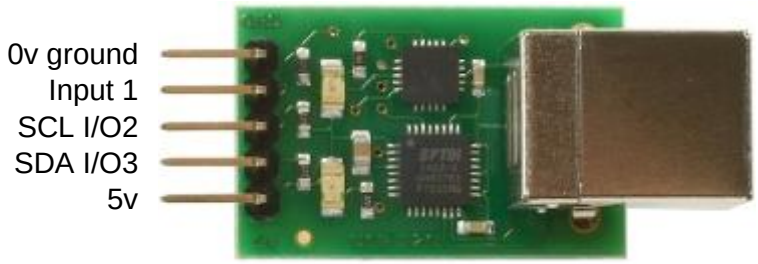
Which COM port?

After installing the drivers, and plugging in the USB-I2C module to a spare USB port, you will want to know which COM port it has been assigned to. This will vary from system to system depending on how many COM ports you currently have installed. To find out where it is, first run "Device Manager" from Windows Settings (for Windows 10) or Control Panel (for Windows 7 or earlier). Now scroll down and open the "Ports (COM & LPT)" tab. You should see the USB serial port listed - COM2 in the example on the right.

NOTE: You need to set PC2FM to the correct COM port in order to control the FM chip using PC2FM.

Connections

The diagram below shows the I2C connections.



See http://www.robot-electronics.co.uk/htm/usb_i2c_tech.htm for more information.