Yani Mustapha

Digital Interfaces for Radio Accessories

Helsinki Metropolia University of Applied Sciences

Degree Programme in Electronics

Bachelor’s Thesis

Date 04.05.2015

Author Title Number of Pages Date	Yani Mustapha Digital Interfaces for Radio Accessories 27 pages + 6 Appendices 04 May 2015
Degree	Bachelor of Engineering
Degree Programme	Degree Programme in Electronics
Instructor	Thierry Baills, Senior Lecturer

This thesis was carried out to develop a concept of how digital interfaces in radio accessory devices could be used. It aims to illustrate the benefits of digital interfaces in regards to radio accessories as well as the drawbacks. Examples of benefits are increase in features of the accessory, such as having embedded sensors, displays, video streaming, better resistance to electromagnetic interference and how fast, easily and reliably the data can be transferred to a radio or phone. Examples of drawbacks are high power consumption and design difficulty.

By means of a Raspberry Pi 2 single-board computer and a camera, a prototype device was made and video was streamed through the USB port to an Android phone. Various interfaces were considered and an explanation about why they were used was given.

The results of the research show that there can be benefits of developing digital interfaces on radio accessories. Various interfaces are available which should be used depending on the needs of the accessory. This usually depends on the type of data that needs to be transferred (audio, video, file transfer or interactive media). Currently the interfaces used in accessories are optimal for the requirements. However if higher data transfer speeds are required to stream video, audio and other data for example, developing an interface such as USB would be beneficial.

Further research was proposed on certain interfaces such as I2C and CAN to determine if it is feasible to stream audio using less wires to radios, the quality of the audio, resistance to EMI and RFI and their respective power consumption.

Keywords	Radio Accessory, Digital Interfaces

Contents

Appendices

Appendix I. Configuration of Raspberry Pi

Appendix II. Camera Serial Interface

Appendix III. Streaming using VLC Player

Appendix IV. Streaming using Gstreamer

Appendix V. USB Tethering between Android Phone and Raspberry Pi

Appendix VI. List of Interface Bit Rates

Abbreviations and Terms

bps – bits per second
Bps – Bytes per second
CAN – Controller Area Network
CPU – Central Processing Unit
EMI – Electromagnetic Interference
FPS- Frames per second
GPIO – General-Purpose Input/Output
I2C – Inter Integrated Circuit
IC – Integrated Circuit
LAN – Local Area Network
LCD – Liquid-Crystal Display
NOOBS – New out of box software
OS – Operating System
PTT – Push to Talk
RAM – Random Access Memory
RFI – Radio Frequency Interference
RPi – Raspberry Pi
SCL – Serial Clock
SD – Secure Digital
SDA – Serial Data
SPI – Serial Peripheral Interface
UART – Universal Asynchronous Receiver/Transmitter
USB – Universal Serial Bus

1 Introduction

The topic of this thesis is digital interfaces that are applicable to a handheld radio accessory. The focus is on the current interface situation which is analog and how digital interfaces can improve radio accessories with an analysis on the feasibility of such implementation.

1.1 Motivation

This thesis was sponsored by a company called Savox Communications Oy that provides safety, rescue and communications products and solutions that improve and save lives, whatever the conditions. The products are aimed to the service of police, security, firefighters, search and rescue teams, military, maritime and industrial markets [1]. The company wanted research on digital accessory interfaces for their handheld professional radio accessories. The reasons being the possible advantages that digital circuitry could bring to a system and more so when digital circuits have been seen to be replacing analog ones in many cases.

The company has had projects where a digital accessory interface was needed and acknowledged that use of digital circuitry might be the future of their products if they are to add features with more demanding data transfer rates.

1.2 Aims for Digital Interfaces

The final goal of the thesis is to research interfaces, their characteristics such as data transfer speed, design complexity and power consumption, properties, requirements and benefits to radio accessories. With this knowledge in mind, it is then possible to choose what kind of interface is needed for a given situation. For example, the company would like to know if using a different interface between their radio accessories and professional hand-held radios could reduce overall cost in production by reducing the number of wires used in the cable. For this thesis, a prototype embedded system device that has advanced digital features, such as video capture and streaming, is made as an example of what sort of digital interface is used in such situations.

2 Theoretical Background

2.1 Radio Accessories

Currently, radio accessories are devices with a remote microphone and speakers which are used as an extension of professional hand-portable radios. The mechanics of these are similar to radios in that there is a push to talk (PTT) button which can be pressed so that a two way communication is possible, while the PTT button is active. Other features that there may be are an emergency button, volume up or down buttons, connections to additional microphone or speaker headset peripherals. The radio accessories are now currently limited to the features that they can have due to their interfaces. While they can transmit audio to the radio, transmitting video is currently not possible since video needs a higher data transfer rate and the professional radios currently do not support video transmission. Being able to determine what kind of interface is ideal for such a feature would be one way of showing the use of digital interfaces.

Figure 1: Savox XG C-C1 handheld radio accessory with two PTT buttons, emergency, volume, mute buttons and a connector for an additional headset [2]

The radio accessory shown in figure 1 was built to work in noisy environments such as in factories, ships or vehicles and is flame resistant. Operating between 3-10V and having a 6mA current drain in its idle state, it does not use a lot of power (18-60mW). Special attention is given to power consumption since the accessories are typically fed from the professional hand-portable radios.

2.2 Modern Communication

As the development of modern communications and computer technologies advances it is important to be able to classify various technologies. Communication itself can be divided into two categories namely:

- Wired communications
- Wireless communications

Wired communications can be further divided into two subcategories:

- Copper wire communications
- Fiber wire communications

Copper wire communications can be categorized into analog and digital communication technologies. Most modern communication technologies use digital data transfer. Generally copper wire communications used in computer technologies are digital technologies, and come in two styles:

- Parallel communications
- Serial communications

Parallel communications can translate data between two devices in multiple bits of data at the same time using several wires. Serial communications transfers data bit-by-bit, by using a single wire (therefore uses less wire) making hardware connections simpler. [3, 1]

2.3 Interfaces

There are many types of interfaces depending on the application. Any two entities that need to communicate with each other need an interface, even in humans. The following interfaces in this chapter illustrate how interfaces work and what are the pros and cons of using them.

2.3.1 Interfaces in Human Beings and Computing

Communication is the transfer of data from one entity to another. The method used to transfer the data however is the Interface. Taking human beings as an example, it could be said that humans have several ways of communicating. Facial expressions, gestures, pointing or using hands, writing, drawing, using equipment such as text messages and computers, touch or eye contact can all be used to “send” information. For the communication to be effective however human beings also have to be able to understand what others are trying to communicate to them. For this they have to understand the language, be able to hear, see, read, feel, have an ability to recognize and use information and memory to recall and understand the information [4].

In computing, an interface is a shared boundary across which two separate components of computer system exchange information. This can consist of both hardware and software. The exchange can be between software, computer hardware, peripheral devices, humans and combinations of these. Some computer hardware devices, such as a touchscreen, can send and receive data through the interface, while others, such as a mouse, microphone or joystick are one way only.

2.3.2 UART

Currently, most of the professional hand-held radios for which the radio accessories are made for support the UART interface. A universal asynchronous receiver/transmitter (UART) is a block of circuitry responsible for implementing serial communication. Essentially, the UART acts as an intermediary between parallel and serial interfaces. On one end of the UART is a bus of eight-or-so data lines (plus some control pins) and on the other is the two serial wires - RX and TX.

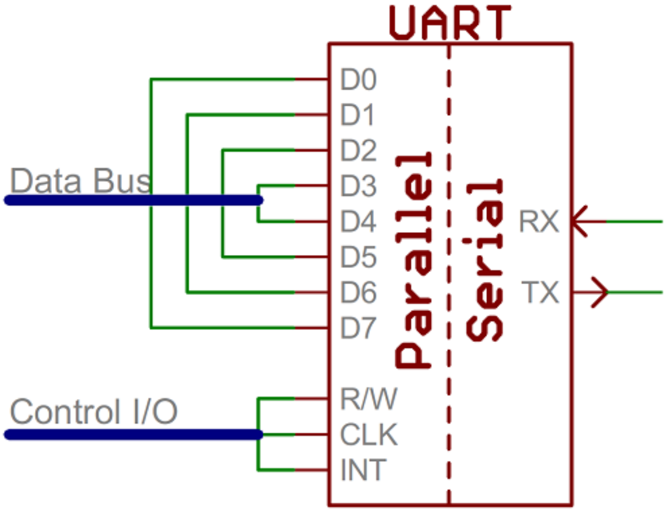


Figure 2: Simplified UART interface. Parallel on one end, serial on the other [5]

UARTs do exist as stand-alone ICs, but they’re more commonly found inside microcontrollers. On the transmit side, a UART must create the data packet - appending sync and parity bits - and send that packet out the TX line with precise timing (according to the set baud rate). On the receive end, the UART has to sample the RX line at rates according to the expected baud rate, pick out the sync bits, and spit out the data [5].

2.3.3 SPI

Serial Peripheral Interface (SPI) is an interface bus commonly used to send data between microcontrollers and small peripherals, such as shift registers, sensors, and SD cards. It uses separate clock and data lines, along with a select line to choose the device to communicate with.

Data and control lines of the SPI and the basic connection:
An SPI protocol specifies 4 signal wires.

1. Master Out Slave In (MOSI) - MOSI signal is generated by Master, recipient is the Slave.
2. Master In Slave Out (MISO) - Slaves generate MISO signals and recipient is the Master.
3. Serial Clock (SCLK or SCK) - SCLK signal is generated by the Master to synchronize data transfers between the master and the slave.
4. Slave Select (SS) from master to Chip Select (CS) of slave - SS signal is generated by Master to select individual slave/peripheral devices. The SS/CS is an active low signal. [6]

There may be other naming conventions such as Serial Data In [SDI] in place of MOSI and Serial Data Out [SDO] for MISO. An example SPI implementation with different naming conventions is shown in figure 2.

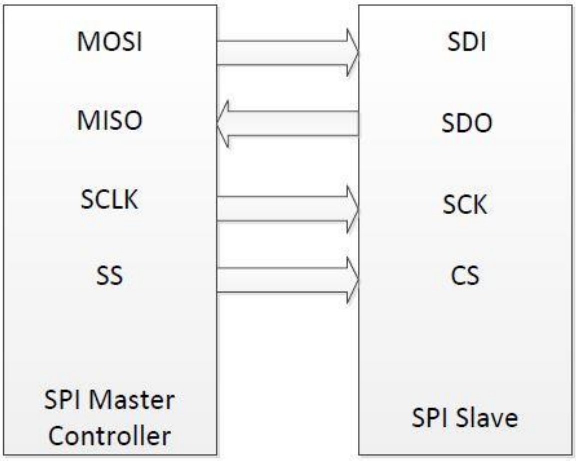


Figure 3: A single master, single slave SPI Implementation.

Among these four logic signals, two of them MOSI & MISO can be grouped as data lines and other two SS & SCLK as control lines.

The usage of these each four pins may depend on the devices. For example, SDI pin may not be present if a device does not require an input (ADC for example), or SDO pin may not be present if a device does not require an output (LCD controllers for example). If a microcontroller only needs to talk to 1 SPI Peripheral or one slave, then the CS pin on that slave may be grounded. With multiple slave devices, an independent SS signal is needed from the master for each slave device [6].

2.3.4 I2C Bus

I2C is a 2-wire, half-duplex, serial bus. The two I2C signals are serial clock (SCL) and serial data (SDA). Both lines are bidirectional and must be connected to VCC via pull-up resistors.

The SCL and SDA pins need to be implemented as an open drain or open collector type to allow for a wired AND function on the bus (”0" wins over "1").

The device initiating data transfers and providing the clock signal on the bus is called a "master". A device being addressed by the master is called a "slave".

The addressed I2C slave can slow down or stop the master by keeping the SCL line pulled low (clock stretching) until it is ready to continue. This way a slow slave device can still keep up with a fast master.

I2C supports multiple masters on the bus through which they are built in bus arbitration. In case of multiple masters trying to send data at the same time, priority is given to the master that first sends a "0" bit [7].

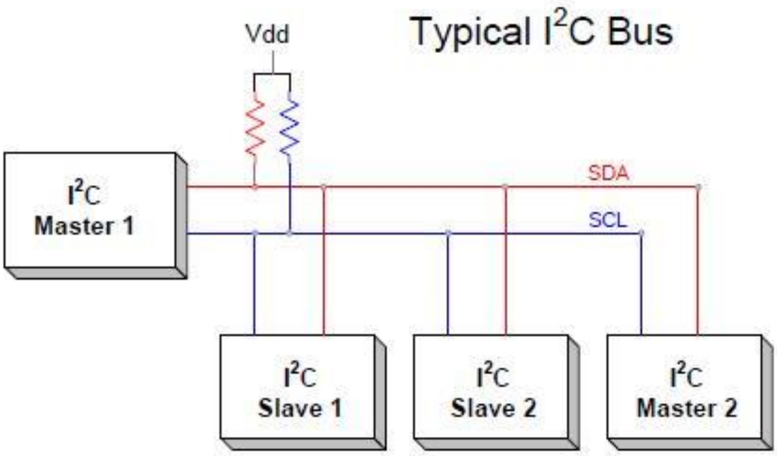


Figure 4: Typical I2C Bus showing the simplicity of the interface hardware wise. [8]

Figure 3 illustrates a typical I2C Bus. The I2C also possesses a complex protocol consisting of start condition, addresses, read/write, Acks, Nacks and Stop condition.

Only four wires are needed for any number of slaves: 2 VCC lines (positive and negative) as well as the Serial Data and Serial Clock lines. Similarly to CAN, the I2C has its own protocol of which a complete understanding is needed for design purposes. Implementation of I2C in radio accessories is discussed in chapter 5.1 (Implementing I2C to Broadcast Audio).

2.3.5 Controller Area Network (CAN)

CAN bus is a standard designed to allow microcontrollers and devices to communicate with each other. It is used mostly in the automotive industry where host computers are not found within a vehicle without a host computer. CAN is very resilient to electromagnetic interference (EMI) and radio frequency interference (RFI) The CAN bus was developed by BOSCH as a multi-master, message broadcast system that specifies a maximum signaling rate of 1 megabit per second (bps). Unlike a traditional network such as USB or Ethernet, CAN does not send large blocks of data point-to-point from node A to node B under the supervision of a central bus master. In a CAN network, many short messages like temperature or RPM are broadcasted to the entire network, which provides for data consistency in every node of the system. [9]

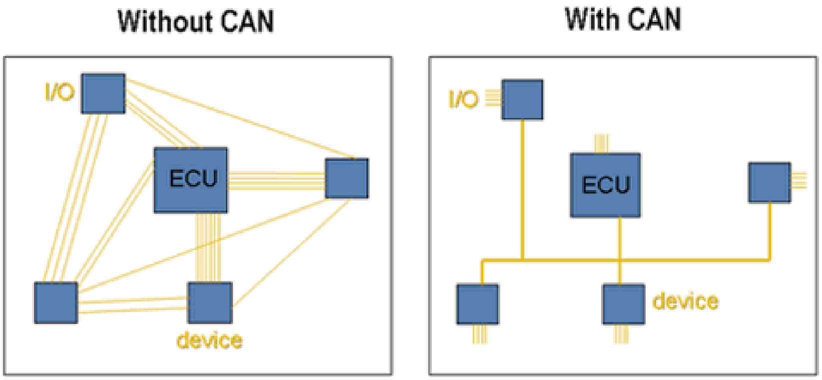


Figure 5: CAN networks significantly reduce wiring. [9]

Figure 4 illustrates a major advantage of CAN; less wiring. The input/output lines, engine control unit (ECU) and other devices are all aware of what is happening in the network.

The CAN standard protocol ISO 11898-1 and ISO 11898-2 or also known as CAN 2.0A or CAN 2.0B define 2 types of message formats. 2.0A defines the CAN standard message which has an 11 bit standard identifier field (or arbitration field) as opposed to the CAN Extended format which has an extra 18 bit identifier field.

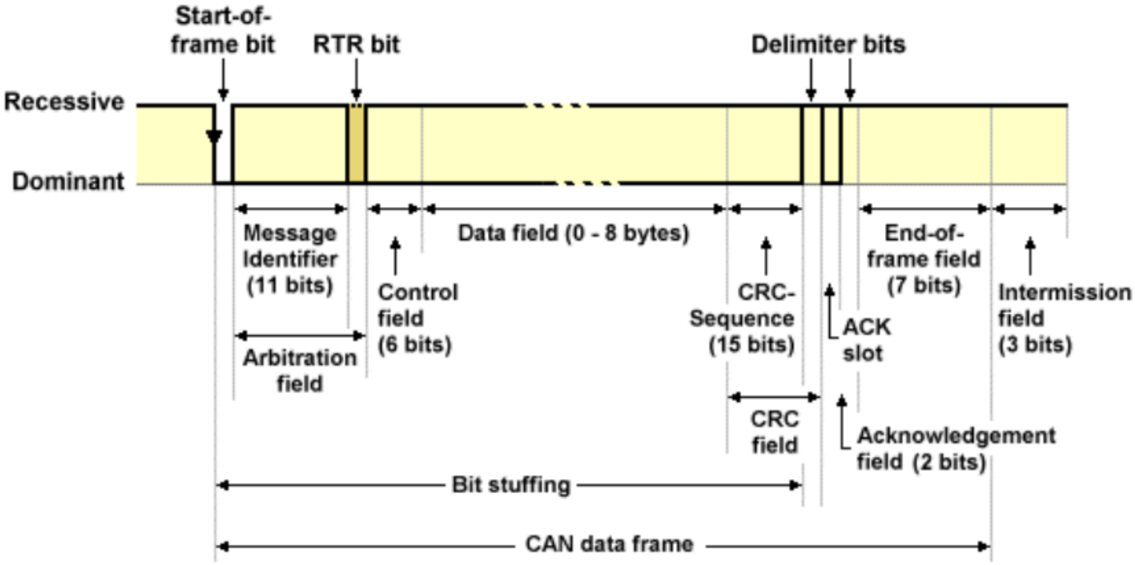


Figure 6: CAN Standard Format showing the complexity of CAM communication

As shown above in figure 5, the protocol requires some training and knowhow to properly get the communication working. Implementation of CAN in radio accessories is discussed in chapter 5.2 (Implementing CAN to Broadcast Audio).

2.3.6 USB

Short for Universal Serial Bus, the USB is an external bus standard which was made to unify various PC connectors into one plug-and-play type of connector supporting hot plugging. The maximum data transfer rate of USB is 12Mbps and it can be used to connect up to 127 peripheral devices such as mice, modems and keyboards.

USB is inexpensive, reliable, provides good performance, and delivers on its promise to provide a ‘plug-and-play’ interface backed by compliance testing. USB is also gaining wide acceptance as the interface of choice for smaller, more portable or mobile consumer electronics devices such as cell phones, digital cameras, PDAs, MP3 players, and more – for data exchange with a PC host. [11]

Figure 6 shows various types of connectors that have been made. USB connectors come in two options: host and peripheral. In the USB standard, there is a difference between the two, and the connectors on cables and devices reflect this.

Figure 7: Types of USB connectors left to right (ruler in centimeters): micro-B plug, UC-E6 proprietary (non-USB) plug, mini-B plug, standard-A receptacle (upside down), standard-A plug, standard-B plug

The new USB type-C (chapter 2.3.9) connectors aim to eliminate the need for different shaped connectors to have only one type of connector. Shape of connectors in radio accessories however greatly depends on the shape dictated by the professional radio companies.

2.3.7 USB 2.0

The motivation for USB 2.0 stems from the fact that PCs have increasingly higher performance and are capable of processing vast amounts of data. At the same time, PC peripherals have added more performance and functionality. User applications such as digital imaging demand a high performance connection between the PC and these increasingly sophisticated peripherals. USB 2.0 addresses this need by adding a third transfer rate of 480 Mb/s to the 12 Mb/s and 1.5 Mb/s originally defined for USB. USB 2.0 is a natural evolution of USB, delivering the desired bandwidth increase while preserving the original motivations for USB and maintaining full compatibility with existing peripherals [12]. USB 2.0, 3.0 and so on are evidence of how interfaces develop based on the current and near future need for faster data transfer speeds.

2.3.8 USB On The Go (OTG)

USB OTG defines two types of configurations: A-devices (devices that have a Standard-A or Mini-A plug inserted), are hosts by default when connected, and B-devices (devices that have a Standard-B or Mini-B plug inserted), are peripherals by default when connected. OTG-devices (formerly known as dual-role-devices) can be either an A-device or B-device, giving it the potential to be either host or peripheral. The status is negotiated between the devices. [13]

Figure 8: OTG enables point-to-point connectivity between peripherals without a traditional PC host. [11]

Figure 7 shows how OTG allows various electronic devices to communicate with each other as well as which device takes the role of host and peripheral.

2.3.9 USB Type-C and USB 3.1

These USB standards offer new characteristics. USB 3.1 has higher data transfer speed characteristics while being backward compatible with USB 3.0 and USB2.0. USB Type-C isn’t the same thing as USB 3.1. USB Type-C is just a connector shape, and the underlying technology could just be USB 2 or USB 3.0.USB. Type-C can be plugged in any polarity, upside-down, solving an age-old issue with USB connectors. USB Type-C also features improved EMI & RFI mitigation and enhanced power delivery options. The development of these standards leads to better characteristics and with USB Type-C, the unification of connectors under only one connector type. [14]

2.3.10 Fiber-Optic Communication

Fiber-optic communication is a way to transfer data from one place to another by sending pulses of light through an optical fiber. The light forms an electromagnetic carrier wave that is modulated to carry information. Researchers at Bell Labs have reached internet speeds of over 100 petabits per second using fiber-optic communication. [15]

Fiber-Optics have numerous advantages such as greatly increased bandwidth and capacity, lower signal attenuation, immunity to electrical noise, immunity to electromagnetic interference and radio-frequency interference, signal security, size and weight and resistance to radiation, temperature variations and corrosion to name a few. However, these come at an expensive price, fragility, risk of being affected by chemicals and require special skills in design and juncture compared to using copper cable.

2.3.11 Wireless Interfaces

Essentially, wireless communication options are simple. They include a modem that provides a defined radio frequency (RF) and an antenna. However, radio waves or signals exhibit very different propagation characteristics depending on their frequency band, so engineers must take into consideration these characteristics when designing a wireless system.

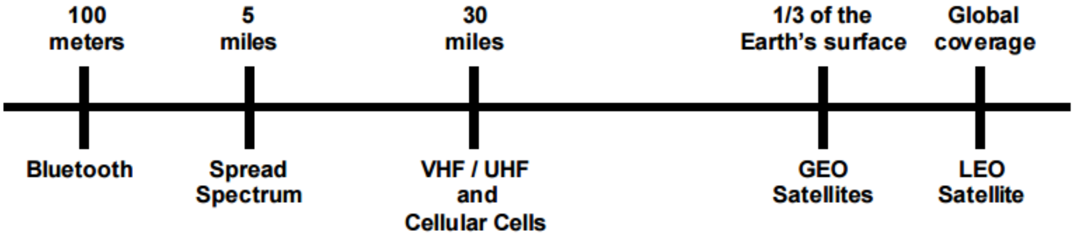


Figure 9: Typical maximum point-to-point transmission range [16, 49]

In addition to frequency band and modem output power, there are other factors to consider, which influence the optimal range of communication. One is “Line of Sight” which means one antenna must “see” the other, without obstructions. Line of sight is not mandatory, but when available, greatly improves performance and range. This is especially true with higher radio frequencies where signal strength is reduced from obstructions such as walls, trees, foliage and concrete and eliminated with metal objects or structures [16, 48].

Positioning, gain, antenna tuning, atmospheric conditions, time of day, ambient frequency, noise, and terrain are also all important variables affecting RF signal and communication range, which is a major risk factor for radio accessories. On top of that, wireless technologies currently have higher power consumption rate than wired technologies and are complex in design. See Appendix VI for a list of wireless interfaces and their respective data transfer rates.

2.3.12 Widely Used Video and Audio Interfaces

As mentioned before, interfaces have a variety of characteristics and properties and therefore serve different purposes. For example the main advantage of USB is its compatibility with very many different computer accessories like mice, keyboards, webcams, portable hard-drives, microphones, printers and so on. Then there are more specific types of cables such as 3.5mm headphone jacks or Digital Optical Audio for audio purposes, VGA or HDMI cables for video purposes, Firewire IEEE 1394 and eSATA cables for fast data transfer or Ethernet cables for networking.

Table 1: This List of Audio and Video Interfaces and their connectors show the multitude of Interface choices available.

Table of Interfaces and their Connectors			
Interface			**Connector example**
Audio or Video	**Digital or Analog**	**Description**	
Audio Only	Analog	PC System Design guide	3.5 mm TRS minijack
		Balanced Audio	6.55 mm TRS audio jack
	Digital	S/PDIF (Sony/Philips Digital Interconnect Format). Via Coaxial or Optical cables	RCA Jack, TOSLINK, BNC
		AES3 (AES/EBU)	RCA Jack, XLR, BNC
Video Only	Analog	Video Graphics Array	D-subminiature 15 pin
		Composite	RCA Jack
		S-Video	Mini-DIN 4 Pin
		Component	3 RCA Jacks
		Composite, S-Video and Component	Vivo/Mini-DIN 9 Pin with breakout cable
	Digital and Analog	Digital Visual Interface	DVI Connector
Video and Audio	Analog	SCART	SCART
	Digital	High-Definition Multimedia Interface	HDMI connector
		DisplayPort	DisplayPort connector
		HDBaseT	8P8C connector
		IEEE 1394 "FireWire"	FireWire or I LINK connectors

In table 1 are video and audio interfaces and examples of the connectors used with them. The connectors play a lesser role as it is the software/interface itself that does most of the work while the connector provides the hardware to support the interface. It can also be noted that all interfaces come in Analog or Digital form. Appendix VI shows a larger list of interfaces and their data transfer rates.

2.3.13 Interfaces used in Radio Accessories

Radio accessories, such as the one in figure 1, currently pass signals in analog format and data through UART interface to communicate to professional hand-held radios. The accessories have several parameters that they need to pass such as Identification which recognizes the device connected, push to talk status, microphone line, speaker outputs and emergency.

In a purely analog accessory, all these interface parameters have their own inputs, outputs, connection wires and they also need a power and ground line to provide power for the accessory to operate. While each parameter has its own connection, the system is robust and has good electromagnetic compatibility properties.

When digital components such as processors are introduced, more features become possible such as making “hands free PTT” or customizing the function of each button. If the whole system were to be digitalized the company would like to know what the new possible features are, whether the data communication between the radio and accessory is unhindered and what is the power consumption.

2.4 Electrical Connectors

For devices to communicate with one another they need connectors. An electrical connector is an electro-mechanical device for joining electrical circuits as an interface using a mechanical assembly. Connectors consist of plugs (male-ended) and jacks (female-ended). The connection may be temporary, as for portable equipment, require a tool for assembly and removal, or serve as a permanent electrical joint between two wires or devices. An adapter can be used to effectively bring together dissimilar connectors. [17]

There are hundreds of types of electrical connectors. Connectors may join two lengths of flexible copper wire or cable, or connect a wire or cable to an electrical terminal. In computing, an electrical connector can also be known as a physical interface.

2.4.1 Properties of Electrical Connectors

Electrical connectors are characterized by their pin-out and physical construction, size, contact resistance, insulation between pins, ruggedness and resistance to vibration, resistance to entry of water or other contaminants, resistance to pressure, reliability, lifetime (number of connect/disconnect operations before failure), and ease of connecting and disconnecting.

They may be keyed to prevent insertion in the wrong orientation, connecting the wrong pins to each other, and have locking mechanisms to ensure that they are fully inserted and cannot work loose or fall out. Some connectors are designed such that certain pins make contact before others when inserted, and break first on disconnection; this protects circuits typically in connectors that apply power for example when connecting safety ground first, and sequencing connections properly in hot swapping applications.

It is usually desirable for a connector to be easy to identify visually, rapid to assemble, require only simple tooling, and be inexpensive. In some cases an equipment manufacturer might choose a connector specifically because it is not compatible with those from other sources, allowing control of what may be connected. No single connector has all the ideal properties; the proliferation of types is a reflection of differing requirements.
Fretting is a common failure mode in electrical connectors that have not been specifically designed to prevent it.

2.5 Types of Information and Their Requirements

To understand what kind of interface should be used in a certain system, the data that is passed and its requirement should be known. Modern digital networks transmit digital information transparently; that is, the network does not necessarily need to know what kind of information the data contain. The information that is transmitted through a network may be any of the following:

- Speech (telephony, fixed, or cellular);
- Moving images (television or video);
- Printed pages or still picture
- Text (electronic mail or short text messaging);
- Music (multimedia messaging);
- All types of computer information such as program files.

However, although all information is coded into digital form, the transmission requirements are highly dependent on the application; because of these different requirements, different networks and technologies are in use.

Table 2: Example of requirements needed for various types of data transfer

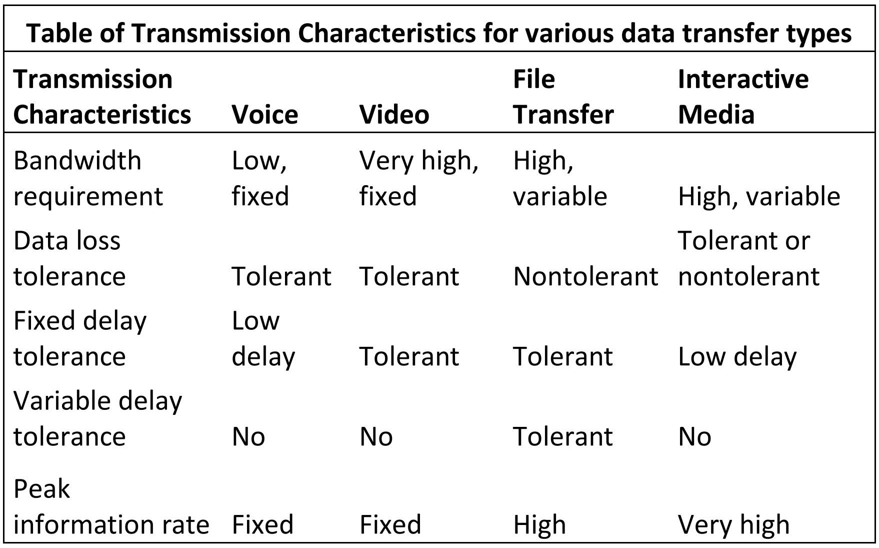


Table of Transmission Characteristics for various data transfer types

Transmission Characteristics	Voice	Video	File Transfer	Interactive Media
Bandwidth requirement	Low, fixed	Very high, fixed	High, variable	High, variable
Data loss tolerance	Tolerant	Tolerant	Nontolerant	Tolerant or nontolerant
Fixed delay tolerance	Low delay	Tolerant	Tolerant	Low delay
Variable delay tolerance	No	No	Tolerant	No
Peak information rate	Fixed	Fixed	High	Very high

Table 2 roughly shows the difference in requirements in different data types. Interfaces should be chosen depending on the type of data transferred.

3 Prototype Design and Implementation

3.1 Hardware Specification

The company sponsoring the thesis, Savox, wished for a concrete example of digital interfaces in the form of video streaming. As shown in table 2, video transfer in general has a very high bandwidth requirement so the hardware chosen would have to reflect the needs. The prototype can be later used in the company as a demonstration system and example of digital radio accessory interface.

3.2 Prototype Structure

USB (On The Go) has been chosen to be the interface for this prototype given that it has the proper data transfer speed characteristics and is the only connector that is supported by the Samsung Galaxy S4 Active phone.

The RPI camera sends raw video to the microcontroller which then has several options available. It can forward the video to the Android phone via USB, save it as a recording on the memory card, display the video to a screen or then do a combination of all three of those options at the same time. Figure 8 shows how the system operates.

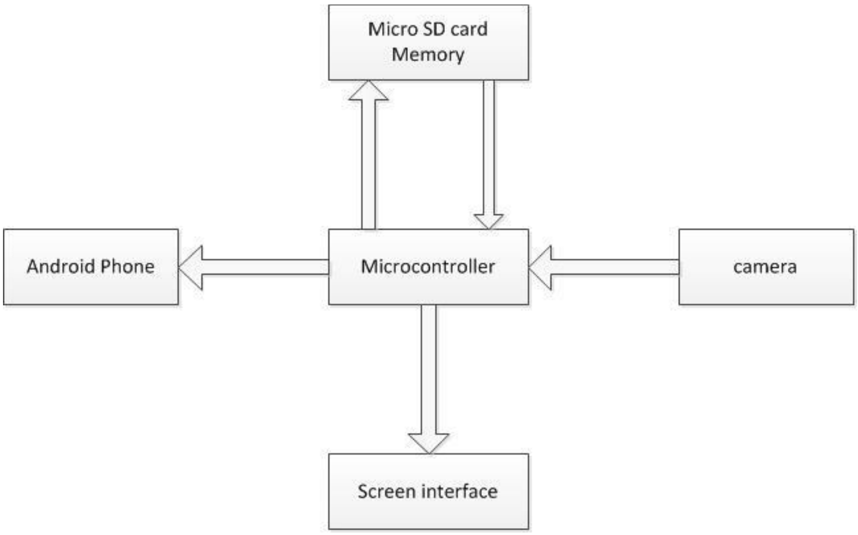


Figure 10: Example of the video streaming device structure

When not streaming video to the Android phone, it should also be possible to retrieve previously recorded videos and display them to the phone or to the screen.

3.3 Choosing the Hardware and Understanding the Interfaces

It is important to understand the interface of each function in the hardware chosen for any project to ensure that they satisfy the requirements of the project itself performance-wise and cost-wise. Various characteristics come into play depending on the application such as power consumption, data transfer rate, connector compatibility, durability, reliability or design complexity.

3.3.1 Microcontroller

A microcontroller is a small computer on a single integrated circuit containing a processor core, memory and programmable input/output peripherals.

A lot of CPU power is needed to be able to encode and stream video while also being given other tasks to do and, as such, a powerful microcontroller such as the Raspberry Pi 2 was chosen. To get a baseline for how much processing speed is needed, the data rate was calculated for the minimum quality video stream wanted. Some of the most interesting characteristics of the RPI 2 are the cheap price, a 900MHz quad-core ARM Cortex-A7 CPU, 1GB of RAM and support for a RPI camera.

The least required resolution set by the company is 640*480 frames of 8-bit black and white video at 25 frames per second (FPS). To calculate how much data transfer per second the device needs, the following formula is given:

$$Datarate = Area\ of\ frame * colorbit\ resolution * Frames\ per\ second \qquad (1)$$

This results in:

$$Datarate = 640 * 480 * 8 * 25 = 61440000\ bytes\ per\ second \qquad (2)$$

This is equivalent to 6.144Mb/s if uncompressed.

H.264 is currently one of the most used video compression formats.

A document called “H.264 Primer” [18] provides a simple formula as a hint to compute the ‘ideal’ compressed output file bitrate, based on the video’s characteristics:

$$Desired\ Bitrate = Area\ of\ Frame * Framerate * motion\ rank * 0.07 \qquad (3)$$

Where the Area of Frame is expressed in pixels and the motion rank is an integer between one and four, one being low motion, two being medium motion, and four being high motion (motion being the amount of image data that is changing between frames.

3.3.2 Camera

For the choice of camera module, the Raspberry Pi Camera Board was the most appropriate choice. It supports Video with a 1080p (1920x1080 pixel) resolution at 30FPS, 720p (1280x720 pixel) at 60FPS and 640x480p at 60 to 90 FPS recording. It is fully compatible with the raspberry pi 2 and not very expensive. See Appendix II to view the Camera Serial Interface.

3.3.3 USB Audio Adapter

This device is needed because the RPI 2 does not have an audio input. When connected via USB and properly configured, the adapter allows the connection of a microphone and speakers.

3.3.4 Memory

The raspberry pi 2 board comes with an integrated SD card reader. An eight gigabyte memory card was the only needed component in this domain. The SD card was formatted using SDformatter4.0 software and contains NOOBS which was installed by downloading and dragging the NOOBS files to the SD card. Approximately 5.7 GB was left free after installation of the OS (Raspbian) on the 8GB SD card.

3.3.5 Screen

Optionally, a screen could be added to the system to display some information. For the company’s accessory applications, the 1.8" Color TFT LCD display with MicroSD Card Breakout, named ST7735R, was chosen.

3.3.6 Battery

Since the Raspberry Pi uses a lot of power, a “USB Battery Pack for Raspberry Pi” was used. At 4400mAh, it provides enough power for a couple of hours of usage.

3.3.7 Phone

A Samsung galaxy S4 Active (with Android operating system version 4.4 installed) was provided by the company for testing purposes. The Raspberry Pi 2 is connected to the Android phone via USB OTG (on the go) through a method called USB tethering. See appendix V for more details about USB tethering.

3.4 Applications Used on the Phone

VLC Player beta and Raspberry Pi Camera Viewer were installed on the phone to view the video stream. Additionally, Raspberry SSH and AndroidVNC were installed. Raspberry SSH enables the user to run commands on the raspberry pi from the phone, using the touch screen as an interface as shown in figure 9. The AndroidVNC allows the scripts on the RPI to be edited easily.

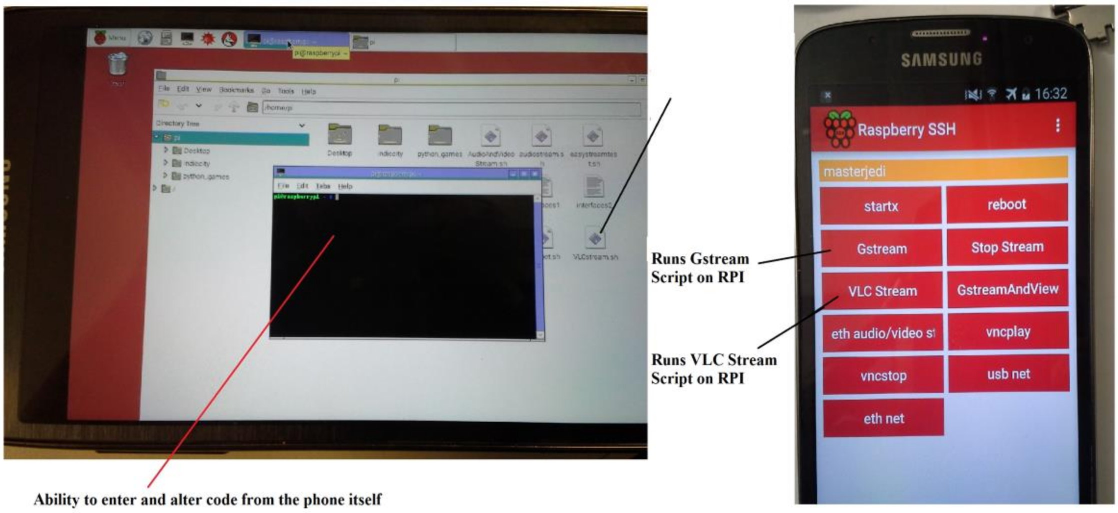


Figure 11: AndroidVNC on the left allows the user to modify and alter the scripts from the phone. On the right is RaspberrySSH, which allows running the scripts on the RPI.

By using software from the “App Store” such as AndroidVNC and RaspberrySSH, the user is able to completely control the RPI without the need for an external keyboard and mouse. Code and Scripts can me added and modified using AndroidVNC while the RasberrySSH can run the scripts such as “VLC Stream”, which starts video streaming via VLC player to the phone, “Stop Stream”, which stops all streaming and “Gstream”, which runs Gstreamer framework and streams the live video to all computers on the Ethernet network as well as to the phone. To view the code for the scripts used in the RPI to stream video, see appendix III for VLC player and appendix IV for Gstreamer.

The screen interface here shows how versatile and advantageous digital interfaces are. The screen provides multiple functions such as editing the software, running the software, viewing the video just by pressing a few buttons on the screen.

4 Results

4.1 The Video Streaming Prototype

The video streaming device was a success in that the Raspberry pi 2 could stream video and audio through various interfaces. It was able to stream video via USB to an android phone with a minimum of 0.4 second delay which with video and for a prototype is an acceptable result. The reason for the 2.6 second delay was mostly due to the time needed for VLC player video encoding. By using Gstreamer framework on the RPi and Raspberry Pi Camera Viewer software on the android phone, the device was able to stream video with less than a 0.4 second delay to the phone.

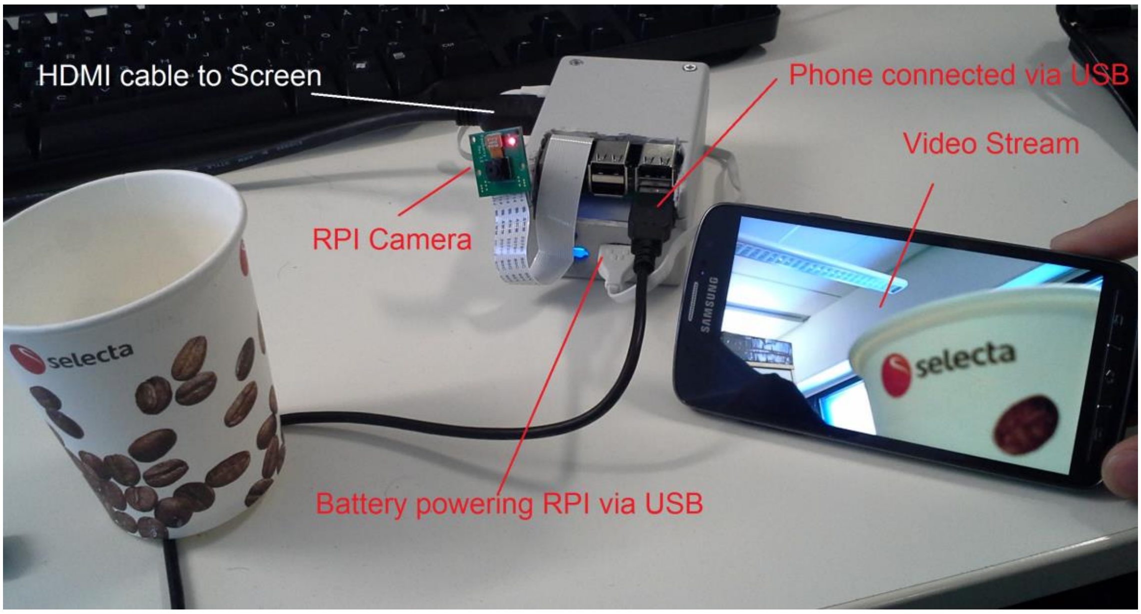


Figure 12: The device streaming to the android phone flawlessly with a 2.6second delay.

As illustrated in figure 10, the device was able to video stream. To view the stream, the RPI was first asked to start “VLC Stream” as shown in figure 9. Once selected, the RPI Camera turns a red LED on indicating that the device is streaming. To view the stream, “VLC player Beta” was opened from the phone and the static ip address of the RPI was given which then opened the live video (See appendix III). The RPI was completely controllable using the Android phone only. To stream with less delay, Raspberry Pi Camera Viewer software was used which uses Gstreamer framework (See appendix IV).

4.1.1 Prototype Performance

Table 3 shows the latency measurements performed on the device while streaming to the android phone via USB and using VLC Player (See appendix III). The latency is the amount of time taken between the capture of the video by the camera and video display on the android phone screen.

Table 3: Latency measurements done between the capture of the video by the raspberry pi camera and the video seen on the Android phone using VLC player

RPI to Android Latency Measurement in Seconds							
Recording Settings	1	2	3	4	5	Average	Standard Deviation
1080p(1920x1080), FPS=30	4.48	4.36	4.45	4.39	4.41	4.42	0.02
1080p(1920x1080) FPS=25	2.67	2.61	2.70	2.69	2.72	2.68	0.02
1080p(1920x1080) FPS=20	2.36	2.47	2.44	2.44	2.60	2.46	0.03
1080p(1920x1080) FPS=15	2.80	2.58	2.60	2.78	2.95	2.74	0.06
1080p(1920x1080), bitrate=3.5mb/s, FPS=30	4.18	4.32	4.43	4.45	4.45	4.37	0.05
1080p(1920x1080), bitrate=7.0mb/s, FPS=30	4.32	4.38	4.36	4.32	4.40	4.29	0.05
640*480 resolution FPS=25	2.77	2.66	2.74	2.71	2.71	2.72	0.02
640*480 resolution FPS=60	14.6	14.6	14.6	14.7	14.6	14.6	0.03
480*320 resolution FPS=25	2.77	2.80	2.80	2.74	2.82	2.79	0.01

From the measurements above done with a stopwatch it can be seen that changing the resolution did not affect the latency at all nor did the bitrate. The only factor changing the average latency were the Frames Per Second (FPS) setting which showed best result at 20 FPS (2.4 second delay).

Using Gstreamer framework (see Appendix IV), it was possible to stream both video and audio through the USB to the android phone and through Ethernet port to other computers on the same network. The delay recorded was difficult to measure being small but was less than 0.4 seconds at maximum resolution (1080p) at 25FPS. To measure small latency, a real time stop watch is recorded via the camera. A picture of the real time stopwatch and the video footage is taken and the time difference between the two stopwatches the one in the video and the real one is the latency.

4.1.2 Prototype Adaptability

The prototype made was a prime example of digital circuitry using multiple interfaces. With 4 USB ports, 40 GPIO (General Purpose Input/Output) pins, Full HDMI port, Ethernet port, combined gives an example of the adaptability of devices that support digital interfaces.

4.1.3 Prototype Reliability

The Raspberry pi 2 takes about 27 seconds to boot up and be ready for use which is not a good feature given the “always ready” requirement of radio accessories. When powered by a constant power source such as a laptop connected to the mains the device was able to stream uninterrupted for 92 hours non-stop. However when powered by the battery, if the battery itself is connected to a power supply to charge while the device is streaming, the RPI reboots causing an interruption of at least 1 minute for the device to be operational which would be unacceptable for a radio accessory.

4.1.4 Prototype Streaming Speed

When streaming, the hardware performance, software performance, network performance and protocol all affect the quality and latency in the streaming. With the values from the previous table it can be seen that there was a 2.6 seconds delay in streaming at very low resolution using VLC. Since the hardware and network performance are very good, the problem lays in encoding/decoding and the protocol.

4.1.5 Prototype Power Consumption

Using the USB battery pack (4400mAh), the device is able to run and stream for a couple of hours. The camera board requires 250mA to operate and the RPI 2 uses 50 mA per core and it has four of them. The RPI2 and camera board therefore use 450mA of current which at 5 volts gives 2.25W of power which is very high compared to radio accessories which currently run on a maximum power of 35mW (see chapter 5.4). The prototype is not optimized from a power consumption point of view and further research on more efficient, less power consuming systems could be done.

4.2 USB OTG Power Consumption

According to the OTG Supplement Rev 1.0, an OTG A-device must provide at least 8mA between 4.4V and 5.25V to power VBUS which gives a minimum of 35mW in power. Devices can negotiate for more current, depending on the host's ability to supply more power. [13]

USB OTG defines low power consumption for portable devices. When there is no active session, VBUS is turned off to save battery power. If the A-device turns off the VBUS, but the B-device wants to use the bus, the B-device can request that the A-device turn on VBUS. This request is called Session Request Protocol (SRP) by the USB OTG Supplement and it is performed by data-line pulsing and VBUS pulsing.

5 Discussion

From the theoretical research done, it is understood that digital interfaces is a broad subject and the ways to apply interfaces to electronics depend largely on the application itself. Each interface type has its advantages, disadvantages, characteristics that make it better suited for a specific area and therefore to use a specific interface, the application and its requirements have to be known.

5.1 Radio Interface Support

For digital interfaces to be implemented onto radio accessories, the radio companies need to provide hardware and software support on their radios for communication to take place as well. This can limit or prevent development of interfaces on radio accessories. One solution to lack of hardware and software support is to use Low-power bridges for I2C or SPI to UART or IrDA or GPIO as shown in figure 13.

Figure 13: A I2C or SPI to UART or IrDA or GPIO bridge circuit [19]

Other types of interface bridge circuits are also available such as USB to UART, USB to I2C and further research on these circuits could be done to determine if they can help radio accessories to communicate to radios and what are the drawbacks.

5.2 Implementing I2C to Broadcast Audio

Local area audio broadcasting systems use analog channels for voice transmission. These are constructed with one main voice transmitter and many loudspeakers. The inherent disadvantages lie in the difficulties to make individual or restricted area communication without additional complex modifications; besides, a pre-planned wiring is necessary, which makes any on-line re-distribution almost impossible. Advantages of a simple I2C voice-broadcasting system are that the network can be built using simply 4 wire lines in series which can potentially lower the cost of wiring in the system. The receiver's ID is changeable for individual or for group reception and the key of encryption can be changed at any time to insure secret communication. Additional devices for control or monitoring can be clipped on the network easily. Even though the original I2C bus was designed as "Inter-Integrated Circuit" for short distance data exchange, however there is no maximum length specified in the I2C specification. [20]

5.3 Implementing CAN to Broadcast Audio

Similar to the idea of implementing I2C to broadcast Audio, broadcasting Audio via CAN is also possible. CAN being used in the vehicle industry because of its high EMC resilience is an eye catcher for radio accessories.

Additional features of broadcasting Audio through CAN are the possibility of data rate up to 1Mbit/s at a bus length of 40 meters. CAN have a sophisticated error detection protocol, where the probability of an undetected error on the bus is very small. Transmission times in CAN are also guaranteed. The performance of the bus is independent of the number of nodes connected to it; CAN is a multi-master bus where the nodes don't have a specific address [21].

5.4 Implementing USB to Radio Accessories

With the high data transfer rate that the USB interface possesses, implementing USB interface on radio accessories would unlock many new possible features such as video streaming or the rapid transfer of largely sized files. It also allows for accessories to be connectable to other devices such as computers, mobile phones and other peripherals. With the launch of USB type-C, future electronic devices, even low in data transfer speed requirement, are likely to adopt USB type-C as the standard, universal connector.

5.5 Implementing a Wireless Interface to Radio Accessories

Due to the complexity in implementation of wireless interfaces, their high power consumption, overall cost and signal loss risk factor, Wireless Interfaces are not deemed a good idea for implementation in radio accessories. However with new technologies such as Wireless USB and IrDA-Giga-IR achieving high data transfer rates (see appendix VI), wireless interfaces could be considered as an additional feature that would provide more options to an accessory.

5.6 Power Consumption of Radio Accessories

Radio accessories currently operate on range of 3.7-7Volts and use between 1-5mA in current for a maximum of 35mW in power. Any implementation of digital interfaces should consider power consumption since that determines how long the device operates or if there is a need for the radio accessory to have its own power supply.

5.7 Professional Radio Connectors

Professional radios nowadays continue to use their own connectors for business related reasons to be incompatible with rival companies. This allows for marketing decisions such as selling customized peripheral devices such as adapters, connectors, microphones, headsets or radio accessories. This however does not prevent professional radios from supporting new technologies.

5.8 Prototype Issues

The prototype made for this thesis underlines a couple of issues with the USB interface, power consumption and booting time. The system rebooting occasionally when another USB peripheral is connected is also another issue. The reliability of the USB interface depends on the system's ability to provide constant voltage and a proper power managing circuit would solve that. More power efficient microcontrollers could be investigated as alternatives for the raspberry pi 2. The Raspberry Pi 2 also contains the Raspbian operating system and software meant for learning which explain the long booting time when turned on.

6 Conclusions

This thesis was set out to investigate about digital interfaces and their application to professional radio accessories. The prototype made using the Raspberry Pi 2 micro-controller and USB as interface was able to show advanced digital capabilities show-casing future possibilities for radio accessories. Certain issues in the prototype such as lowering power consumption and having better reliability using USB as interface could be further investigated. The USB type-C interface with good data transfer characteristics is expected to become the standard interface for communication between devices for all data transfer types. Examples of how interfaces could be applied to radio accessories in different ways were given and explained such as implementing a USB interface to transfer video, audio and other data types as well as becoming compatible with more devices such as smartphones. SPI, I2C or CAN were investigated as an alternate method of broadcast audio and could be further investigated to show their feasibility in regards of power consumption, design difficulty and production costs but would not be a major improvement overall. Wireless interfaces were also debated and deemed unacceptable to be used as sole interface in radio accessories due to several factors. Radio accessories depend greatly on the interfaces supported by the professional hand-held radios and therefore any implementation of digital interface on the accessories will have to be mutual with that on the radios.

References

[1] Savox Communications, Who We Are [online]. URL: http://www.savox.com/company-details/. [Accessed 07 April 2015]

[2] XG C-C1 - software configurable RSM with superior audio quality [online] URL: www.savox.com/products/xg-c-c1/. [Accessed 04 May 2015]

[3] Ying Bai, The Windows Serial Port Programming Handbook, The Fundamentals of Serial Port Communications. ISBN 0-8493-2213-8

[4] Stroke4carers, Communication Strategies [online] URL: http://www.stroke4carers.org/?p=280. [Accessed 07 April 2015].

[5] Sparkfun, Serial Communications, UART. URL: https://learn.sparkfun.com/tutorials/serial-communication/uarts. [Accessed 04 May 2015]

[6] Freescale Semiconductors, SPI Block Guide V03 06 21 January 2000 [online] URL: http://www.ee.nmt.edu/~teare/ee308l/datasheets/S12SPIV3.pdf. [Accessed 19 April 2015]

[7] Jean-Marc Irazabal, Steve Blozis, Philips Semiconductors, I2C Manual, AN10216-01, 24 March 2003

[8] Hobbizine, Adding memory with i2c eeproms [online] URL: http://picaxe.hobbizine.com/eeprom.html. [Accessed 04/05/2015]

[9] Steve Corrigan, Introduction to the Controller Area Network (CAN). Application Report SLOA101A–August 2002–Revised July 2008 [online] URL: http://www.ti.com/lit/an/sloa101a/sloa101a.pdf. [Accessed 10 April 2015].

[10] National Instruments, Controller Area Network (CAN) Overview, 01 August, 2014

[11] Universal Serial Bus Specification, revision 2.0, 27 April 2000.

[12] USB Specification Expanding, Boosting Performance Up to 40 Times beyond Current Capability [Online] URL: http://www.usb.org/developers/usb20/backgrounder/. [Accessed 20 April 2015]

[13] On-The-Go and Embedded Host Supplement to the USB Revision 2.0 Specification, Revision 2.0 version 1.1a, 27 July 2012.

[14] Universal Serial Bus Specification, Revision 3, 11 August 2014

[15] Scott South, WaterWorld, January 2005, Environmental Monitoring: Guide to Selecting Wireless Communication Solutions

[16] Kush Amerasinghe, H.264 for the rest of us. [Online] URL: http://www.adobe.com/content/dam/Adobe/en/devnet/video/articles/h264_primer/h264_primer.pdf. [Accessed 04 May 2015]

[17] Bell Labs breaks optical transmission record, 100 Petabit per second kilometer barrier, Phys.org. 29 September 2009.

[18] Robert S. Mroczkowski, Electrical Connector Handbook Theory and Applications, McGraw Hill, 1998 ISBN 0-07-041401-7

[19] NXP, I2C/SPI to UART/IrDA/GPIO Bridge ICs [online] URL: http://www.nxp.com/products/interface_and_connectivity/bridges/i2c_s
slave_to_uart_irda_gpio_bridges/. [Accessed 04 May 2015]

[20] Chao-Huang Wei, Yun-Chung Lin, Implementation of Local Area Digital Audio Broadcasting System upon I2C Network, IEEE Xplore 2004.

[21] John Cosgrove, Audio Signals Over The Controller Area Network (CAN), University of Bath, 09/2004.

[22] Mipi, Camera Serial Interface [online]. URL: http://mipi.org/specifications/camera-interface. [Accessed 04 May 2015]

[24] Adam. Streaming Video Using Gstreamer [online]. URL: http://www.raspberry-projects.com/pi/pi-hardware/raspberry-pi-camera/streaming-video-using-gstreamer. [Accessed 04 May 2015]

Appendices

Appendix I

Configuring the Raspberry Pi

Step 1: Download “NOOBS” from www.raspberrypi.org/downloads/

Step 2: Format micro SD card (with SD formatter 4 for example).

Step 3: Copy/paste contents of NOOBS (latest version) on SD card.

Step 4: Insert SD card in RPi <- Raspberry Pi

Step 5: Install Raspbian, this generally takes a long while (about 10 minutes)

Step 6: Raspberry Pi software configuration tool opens up after installation

Step 7: (optional): Enable Camera, Set time zone etc…

Step 8: Reboot

Step 9: Login with “pi” as username and “raspberry” as password

Step 10: The Raspberry pi is now ready to be used! However it is best to update and upgrade all the packages first.

Step 11 (optional): To reconfigure the keyboard language or layout type in the terminal

Step 12: Make sure a mouse, keyboard and Ethernet cable are connected. Internet is needed to update and download software. Using internet from a wireless dongle or via USB from a smartphone is also possible.

Step13: Update Raspbian by typing in terminal

```
sudo apt-get update
```

Step 14: Upgrade all raspbian packages by typing in terminal

```
sudo apt-get upgrade
```

This will prompt the user with a Yes or No question which the user answers by pressing “Y“ for yes or “N” for no and the enter key.

Step 15 (optional): Typing “`startx`” in the command terminal starts the graphical user interface.

Step 16 (optional): If the internet doesn’t work through the Ethernet cable right away

Type in the terminal:

```
sudo nano /etc/network/interfaces
```

In general it should look like this:

```
Auto lo
Auto eth0
iface lo inet loopback
```

```
iface eth0 inet dhcp
allow-hotplug wlan0
iface wlan 0 inet manual
wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
iface default inet dhcp
```

Restart the internet settings by rebooting or typing

```
          sudo /etc/init.d/networking restart
```

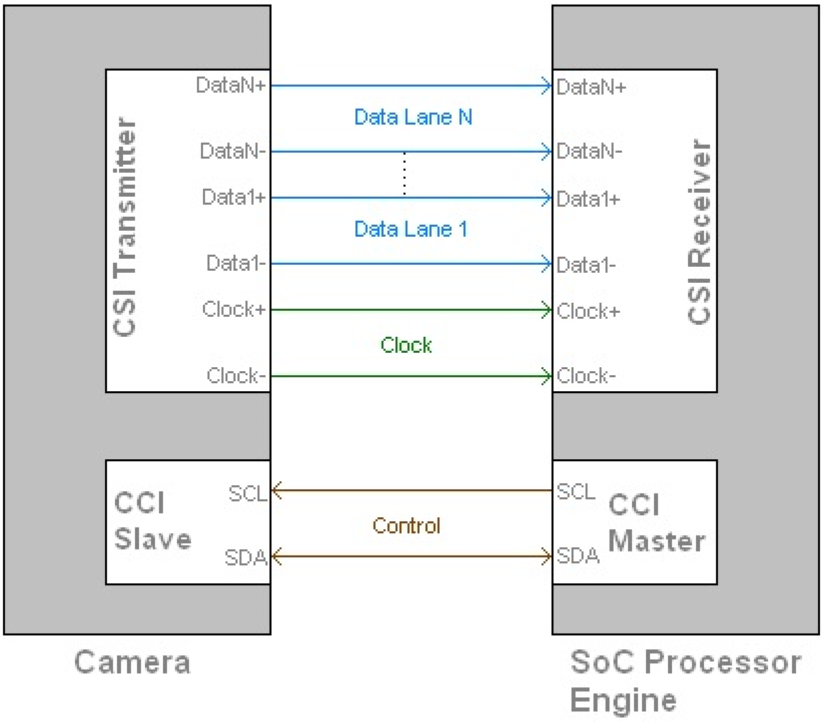
Step 18

Now the Raspbian is updated, upgraded and all the tools needed to make software and have fun are installed.

Appendix II

Camera Serial Interface

The camera ribbon cable is connected to the CSI (Camera serial interface) on the RPi.



The Raspberry Pi CSI connector is a surface mount ZIF 15 socket, used for interfacing a camera through a ribbon cable. The CSI-2 specifications are very detailed and describe the physical layer known as D-PHY2. The signaling scheme of this physical layer, known as Low Voltage Differential Signaling (SubLVDS), is a modified version of the IEEE1596.3 LVDS specification. It is a system for low voltage 1.2 V applications, allowing data rates of up to 800 Mbps per lane with 1 Gbps set as a practical limit. In practice, the data rate can vary a lot and depends upon the quality of the interconnections. A maximum of four physical data lanes are allowable in this specification, however two are available for the Raspberry Pi

This is a high-speed data communication bus and noise is of huge concern to the design engineer. Although this type of serial communication generates negligible crosstalk, the specification suggests using minimum clock rates for the camera module. The CSI transmission clock is source synchronous and the main processor may produce it instead to avoid noise interference on the camera module. The data transmission supports a wide range of data types such as RGB, RAW, YUV, generic, or byte based programmer defined. [22]

For a 1 Gb bandwidth, the specification defines six pins consisting of two for data, two for control, and two for clock. For 2 Gb bandwidth, eight pins consisting of four for data (two lanes), two for control, and two for clock are required. There is also an additional pair of data lines for serial communication to set the bridge registers.

However, the recently announced CSI-3 interface offers many features which permit low-power, high-performance camera sensor integration. The CSI-3 standard follows the very successful CSI-2 standard that is forecast by InStat to reach 70% penetration of mobile and consumer electronic devices with image sensors by 2016. A more detailed breakdown of the architectures and communication protocols of these standards is useful for understanding how and where to apply them.

Appendix III

Streaming using VLC player.

VLC (beta) is installed on the android phone through the google play store.

Installing VLC Player using the command terminal:

```
sudo apt-get install vlc
```

Streaming The Camera Video Using RTSP

Enter the following command to start the streaming:

```
raspivid -o - -t 0 -n | cvlc -vvv stream:///dev/stdin --sout
'#rtp{sdp=rtsp://:8554/}' :demux=h264
```

raspivid is used to capture the video

"`-o -`" causes the output to be written to stdout

"`-t 0`" sets the timeout to disabled

"`-n`" stops the video being previewed (remove if there is a need to see the video on the HDMI output)

`cvlc` is the console vlc player

"`-vvv`" and its argument specifies where to get the stream from

"`-sout`" and its argument specifies where to output it to

Appendix IV

Streaming using Gstreamer

The GStreamer core function is to provide a framework for plugins, data flow and media type handling/negotiation. It also provides an API to write applications using the various plugins. The framework is based on plugins that will provide the various codec and other functionality. The plugins can be linked and arranged in a pipeline. This pipeline defines the flow of the data. Pipelines can also be edited with a GUI editor and saved as XML so that pipeline libraries can be made with a minimum of effort.

Gstreamer is installed by default on the raspbian but however if it is not installed or is not detected, these two lines usually ensure that it is properly installed:

```
sudo get-apt remove gstreamer
sudp apt-get install gstreamer
```

To start streaming on the raspberry pi using Gstreamer, the following code is used [24]:

```
raspivid –t 0 –h 720 –w 1080 –FPS 25 –hf –b 6000000 –o - | gst-launch-1.0 –v fdsrc ! h264parse! rtph264pay config-interval=1 pt=96 ! gdppay ! tcpserversink host=RPIaddress port=5000
```

To view the stream on a computer first Gstreamer is installed. Then a batch file such as “stream.bat” is created using a text editor and the following code is added into it.

```
@echo off

C:\gstreamer\1.0\x86_64\bin\gst-launch-1.0 -v tcpclientsrc host=RPIaddress port=5000 ! gdpdepay ! rtph264depay ! avdec_h264 ! videoconvert ! autovideosink sync=false

PAUSE
```

Appendix V

USB Tethering between Android Phone and Raspberry Pi

For tethering to occur, Oh the android phone, set Tethering on in Settings

On raspberry pi,
Typing `ifconfig` in the command terminal will reveal which USB device is the phone (example: usb0). The following code is typed to perform tethering from the RPI.

```
sudo dhclient usb0
```

Typing ifconfig again will reveal the usb0 ip address.
It is then possible to open the streaming video straight from the USB by inserting the usb0 ip address and the port number set in appendix III or IV. The following command is entered using a rstp compatible software:
“`rstp://192.168.xx.xxx:portnumber/`”

Appendix VI

List of Interface bit rates

Table 5 and 6 were found on Wikipedia and might contain erroneous information as they currently lack any reference about the information presented. For this reason, the reference is listed here instead of on the reference list.

http://en.wikipedia.org/wiki/List_of_device_bit_rates. [Accessed 04 May 2015]

Table 4: List of peripherals and their respective data transfer rates

Technology	Rate (bit/s)	Rate (byte/s)	Year
CBM Bus	2.7 kbit/s	0.34 kB/s	1981
Apple Desktop Bus	10.0 kbit/s	1.25 kB/s	1986
Serial MIDI	31.25 kbit/s	3.9 kB/s	1983
Serial EIA-232 max.	230.4 kbit/s	28.8 kB/s	1962
Serial DMX512A	250.0 kbit/s	31.25 kB/s	1998
Parallel (Centronics)	1 Mbit/s	125 kB/s	1970 (Standardised 1994)
Serial 16550 UART max.	1.5 Mbit/s	187.5 kB/s	
USB 1.1 ("Low-Bandwidth")	1.536 Mbit/s	192 kB/s	1996
Serial UART max	2.7648 Mbit/s	345.6 kB/s	
GPIB/HPIB (IEEE-488.1) IEEE-488 max.	8 Mbit/s	1 MB/s	late 1960s (Standardised 1976)
Serial EIA-422 max.	10 Mbit/s	1.25 MB/s	
USB 1.1 ("Full-Bandwidth")	12 Mbit/s	1.5 MB/s	1996
Parallel (Centronics) EPP 2 MHz	16 Mbit/s	2 MB/s	1992
Serial EIA-485 max.	35 Mbit/s	4.375 MB/s	
GPIB/HPIB (IEEE-488.1-2003) IEEE-488 max.	64 Mbit/s	8 MB/s	
FireWire (IEEE 1394) 100	98.304 Mbit/s	12.288 MB/s	1995
FireWire (IEEE 1394) 200	196.608 Mbit/s	24.576 MB/s	1995
FireWire (IEEE 1394) 400	393.216 Mbit/s	49.152 MB/s	1995
USB 2.0 ("Hi-Speed")	480 Mbit/s	60 MB/s	2000
FireWire (IEEE 1394b) 800	786.432 Mbit/s	98.304 MB/s	2002
Fibre Channel 1 Gb SCSI	1062.5 Mbit/s	100 MB/s	
FireWire (IEEE 1394b) 1600	1.573 Gbit/s	196.6 MB/s	2007
Fibre Channel 2 Gb SCSI	2125 Mbit/s	200 MB/s	
eSATA (SATA 300)	3 Gbit/s	375 MB/s	2004

CoaXPress Base (up and down bidirectional link)	3.125 Gbit/s + 20.833 Mbit/s	390 MB/s	2009
FireWire (IEEE 1394b) 3200	3145.7 Mbit/s	393.216 MB/s	2007
External PCI Express 2.0 ×1	4 Gbit/s	500 MB/s	
Fibre Channel 4 Gb SCSI	4.25 Gbit/s	531.25 MB/s	
USB 3.0 ("SuperSpeed")	5 Gbit/s	625 MB/s	2010
eSATA (SATA 600)	6 Gbit/s	750 MB/s	2011
CoaXPress full (up and down bidirectional link)	6.25 Gbit/s + 20.833 Mbit/s	781 MB/s	2009
External PCI Express 2.0 ×2	8 Gbit/s	1000 MB/s	
USB 3.1 ("SuperSpeed+")	10 Gbit/s	1250 MB/s	2013
Thunderbolt	10 Gbit/s × 2	1250 MB/s × 2	2011
External PCI Express 2.0 ×4	16 Gbit/s	2000 MB/s	
Thunderbolt 2	20 Gbit/s	2500 MB/s	2013
External PCI Express 2.0 ×8	32 Gbit/s	4000 MB/s	
External PCI Express 2.0 ×16	64 Gbit/s	8000 MB/s	

Table 5: List of Wireless Personal Area Networks and their Data Transfer Speeds

Technology	Rate (bit/s)	Rate (byte/s)	Year
ANT	20 kbit/s	2.5 kB/s	
IrDA-Control	72 kbit/s	9 kB/s	
IrDA-SIR	115.2 kbit/s	14 kB/s	
802.15.4 (2.4 GHz)	250 kbit/s	31.25 kB/s	
Bluetooth 1.1	1 Mbit/s	125 kB/s	2002
Bluetooth 2.0+EDR	3 Mbit/s	375 kB/s	2004
IrDA-FIR	4 Mbit/s	500 kB/s	
IrDA-VFIR	16 Mbit/s	2 MB/s	
Bluetooth 3.0	24 Mbit/s	3 MB/s	2009
Bluetooth 4.0	24 Mbit/s	3 MB/s	2010
IrDA-UFIR	96 Mbit/s	12 MB/s	
WUSB-UWB	480 Mbit/s	60 MB/s	
IrDA-Giga-IR	1024 Mbit/s	128 MB/s	