Wireless Home Automation using IoT (Internet of Things)

submitted by

Shubham Raj

Enrollment No. 15114069

Kamal Kant

Enrollment No. 15114034

Under the guidance of

Dr. Sudip Roy

Assistant Professor

Department of Computer Science and Engineering

Department of Computer Science and Engineering

Indian Institute of Technology Roorkee

Roorkee, Uttarakhand- 247667, INDIA

January 2,2017

TABLE OF CONTENTS:

Page No.

Abstract

IoT or Internet of Things is an upcoming technology that allows us to control hardware devices through the internet. Here we propose to use IoT in order to control home appliances, thus automating modern homes through Internet. This system uses two loads to demonstrate as house lighting and a fan. Our user friendly interface allows user to easily control these home appliances through Internet. For this system, we use Raspberry Pi Model 3B. This Raspberry Pi is interfaced with a Wi-Fi modem to get user commands over internet. Relays are used to switch loads. The entire system is powered by 220V ac input, which is converted into 5V dc input for Raspberry Pi. After receiving user command over internet, Raspberry processes these instructions to operate these loads accordingly and produce desired outcome. Thus the system allows for efficient home automation over the internet.

Keywords: Wireless Home Automation system, IoT, Raspberry Pi

Wireless Home Automation using IoT (Internet of Things)

The process of controlling or operating various equipment, machinery, industrial processes, and other applications using various control systems also with less or no human intervention is termed as automation. There are various types of automation based on the application they can be categorized as home automation, industrial automation, autonomous automation, building automation, etc... Our project concerns about wireless home automation using IoT (Internet of Things)

Home Automation

Home automaton is the process of controlling home appliances automatically using various control system techniques. The electrical and electronic appliances in the home such as

fan, lights, outdoor lights, fire alarm, kitchen timer, etc., can be controlled using various control techniques.

There are various techniques to control home appliances such as IoT based home automation over cloud, home automation under Wi-Fi through android apps from any smartphone, Raspberry based home automation, home automation by android application based remote control, using digital control, RF based home automation system and touch screen based home automation. Wireless home automation using IoT is an innovative application of internet of things developed to control home appliances remotely over the cloud.

Required Components and Materials

Hardware Specifications

- Raspberry Pi 3B
- Relay board
- Jumper wires
- Wi-Fi Modem
- Resistors
- Loads (home appliances)

Software Specifications

- Android Studio
- Raspbian OS
- Node.js, html programming languages

1. Raspberry Pi

The Raspberry Pi is a series of credit card–sized single-board computers developed in the United Kingdom by the Raspberry Pi Foundation with the intention of promoting the teaching of basic computer science. Raspberry Pi is an open source hardware technology combined with a programming language and an Integrated Development Environment

(IDE). C++ uses for programming Arduino. HTSQL (Hyper Text Structured Query Language) to provide a web interface to database that is easy to query via the web browser.

Operating systems that are available to install the Raspberry is Raspbian.

2. Relay

   A Relay is electrically operated switches, which allow low power circuits to switch a relatively high voltage or current on/off. For a relay to operate a suitable pull in and holding current should be passed through its coil. Relay coils are designed to operate from a particular voltage often its 5V or 12V. The function of relay driver circuit is to provide the necessary current energize the relay coil, when a LOGIC 1 is written on the PORT PIN thus turning on the relay. The relay is turn off by writing LOGIC 0 on the port pin. In our system four relays are used for device control.

3. Jumper wire

   A jump wire is an electrical wire or group of them in a cable with a connector or pin at each end which is normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering.

4. Wi-Fi Modem

   A wireless router is a device that performs the functions of a router and also includes the functions of a wireless access point. It is used to provide access to the Internet or a private computer network. It can function in a wired LAN (local area network), in a

wireless-only LAN (WLAN), or in a mixed wired/wireless network, depending on the manufacturer and model.

5. Resistor

A resistor is a passive two-terminal electrical component that implements electrical resistance as a circuit element. In electronic circuits, resistors are used to reduce current flow, adjust signal levels, to divide voltages, bias active elements, and terminate transmission lines.

6. Loads

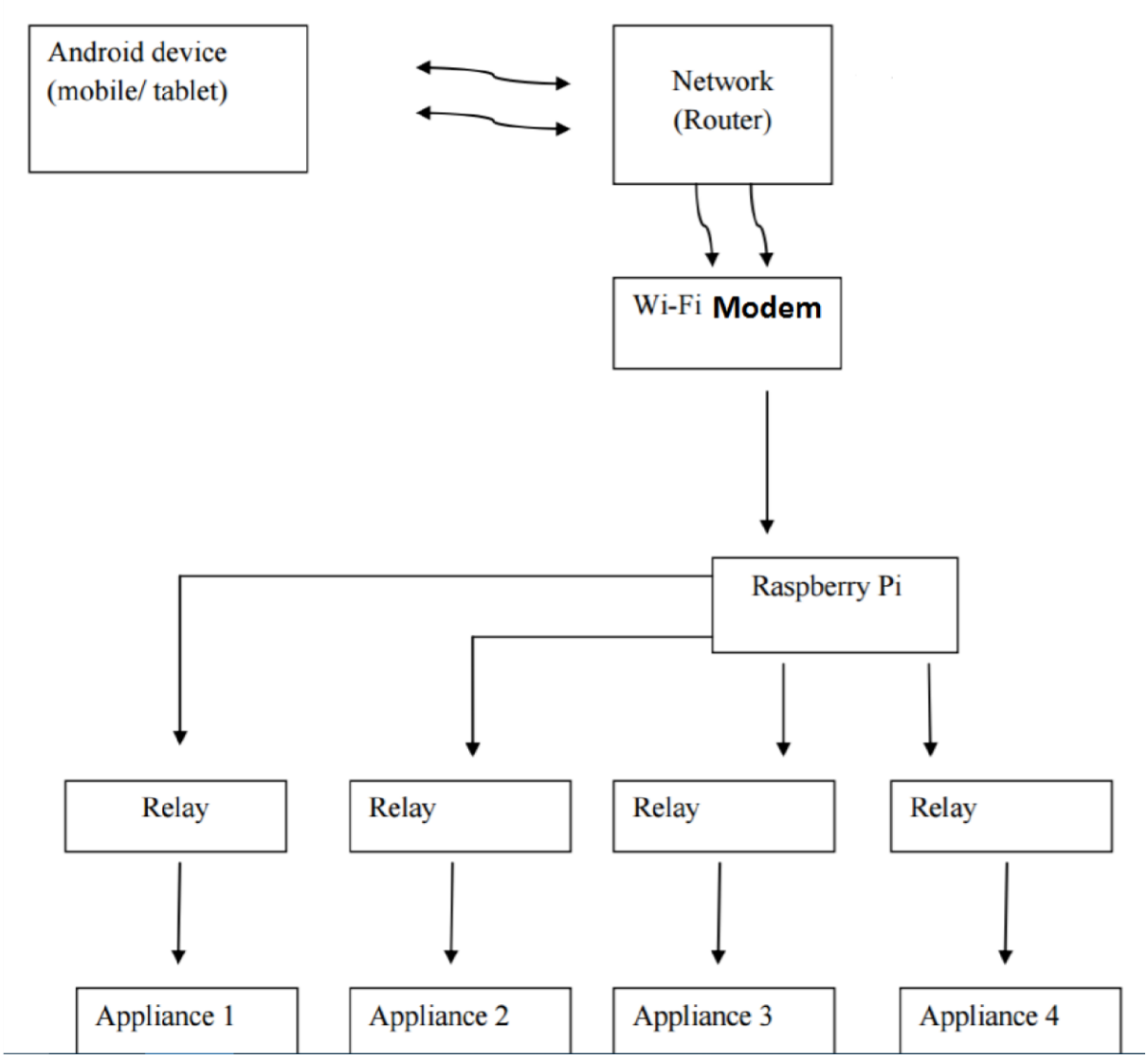
Loads are usual home appliances like bulbs and fans, used here for the purpose to show the working of project.

Proposed Work:

Project Objectives:

Android controlled Smart Home Automation should be able to control the home appliances wirelessly with effectively and efficiently.

Block diagram of proposed system



Technology Exposures That Project Provides:

1. Google's Android open source technology.

2. Wi-Fi technology.

3. Interfacing Wireless Adapter to Raspberry pi.

4. Interfacing relays with ac and dc power sources.

5. Using Relays as a Switch.

6. Embedded programming

Scopes:

The project aims at designing a prototype for controlling the home appliances that can be controlled wirelessly via an application that provides the features of switching ON/OFF the relays. An application is run on android device. The system can be used in wide range of areas.

The system integrated with different features can be applied in the following fields.

• **The system can be used in home, small offices to the big malls**

The system can be used from home to offices to control the electrical appliances.

• For remote access of appliances in internet or intranet.

The home/office appliances can be controlled in intra-network or can be accessed via internet.

• For the development of technology friendly environment

The system incorporates the use of technology and making smart home automation. By the use of day to day gadgets we can utilize them for different prospective.

Project Management

This project constituted development of application as its major part as well as the hardware to control home appliances. Management of any project has several steps or processes in it. So, our projects can be described under the following steps-:

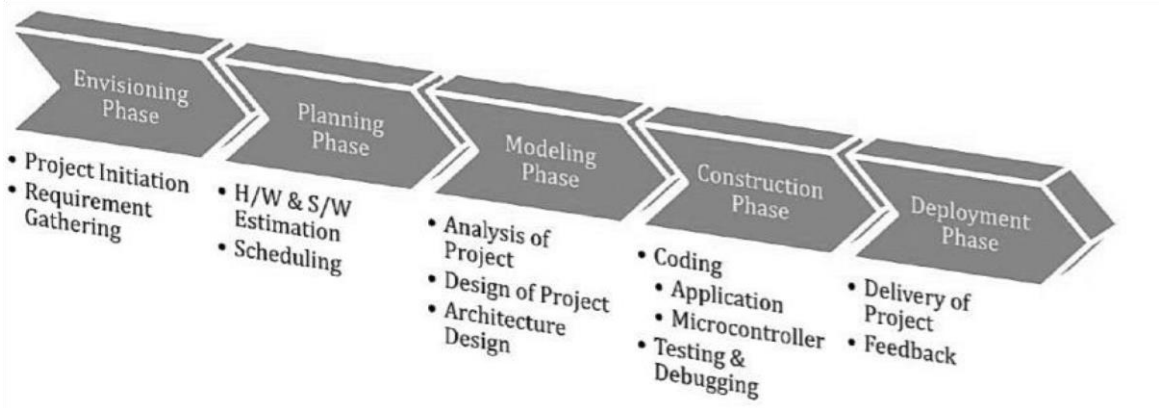


Figure 1: Waterfall Model

Experimentation

In this step, we were discussing about the necessary equipment and materials. We were studying about the similar projects, gathering the information of programming language to be used. We were developing simple algorithms and flowcharts.

Design

In this phase, we were designing the layout of the application. The necessary features to be included. We were designing the power strip to connect the home appliances that can be controlled via GPIO pins.

Development and Testing

In this phase, the development of application was performed. The bugs were identified and removed. We consulted many software experts for the evaluation of our application. Hardware design includes the design of power strip.

Real-World Testing

Finally, our system was ready to be tested in the real electrical appliances.

Project Description / Modelling Work

Hardware Environment

- Raspberry Pi as the controller for its processing power and large developer community.
- 3 relays are connected to power strips.
- GPIO pins are connected to relays. Relays are used as switch.

Software Environment

- RPI-GPIO library
- Android Studio
- Raspbian OS
- Node JS
- Git Command line / Git terminal
- Putty (SSH terminal for remote connection with the Raspberry Pi)

Codes for the Project: -

Server code: Index.js (server created using node js)

```
        Var
gpio=require("gpio");
            var express= require('express');    var app=express();
            var http= require('http').Server(app);     var io= require('socket.io')(http);
            app.use(express.static(__dirname + '/public'));
            app.set('port',(process.env.PORT||3000));
            http.listen(app.get('port'),function(){      console.log("listening to port
      number"+app.get('port'));
            });
            var off1,off2;
            var pyshell=require('python-shell');
            app.get('/',function(req,res){        res.sendFile(__dirname +
      '/index.html');
            });
            io.sockets.on('connection', function (socket) {



            //light1onoff code


            socket.on('light1',function(data){ console.log('working');



                  pyshell.run('a.py',function(err){
                                                                  if(err) throw
            err;
                                                 });




                                                });


            socket.on('light1off',function(data){console.log('off');


                        pyshell.run('b.py',function(err){
                                                                  if(err) throw
```

```
err;
                                        });

                                        });


    //fan1onoff code


    socket.on('fan1',function(data){ console.log('working');


    pyshell.run('e.py',function(err){
                                                        if(err) throw
    err;
                                        });




                                    });


socket.on('fan1off',function(data){console.log('off');
                                        pyshell.run('f.py',function(err){
                                                        if(err) throw
            err;
                                                    });


                                                    });


    //light2onoff code


socket.on('light2',function(data){ console.log('working');
```

```
pyshell.run('c.py',function(err){
  if(err) throw
err;
        });

        });

socket.on('light2off',function(data){console.log('off');
    pyshell.run('d.py',function(err){
      if(err) throw
err;
        });

        });
    });
```

Python script to control the GPIO pins to turn ON / OFF appliances:

```
import
RPi.GPIO as
GPIO
import time

GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(4,GPIO.OUT)

state=True
    GPIO.output(4,False)
```

For Complete Code (Android Studio/ User Interface and other) refer to:

1. **https://github.com/raj808569/homeautomationIOT**
2. **https://github.com/kmlkant9/RaspberryPi**

Software Description

Java:

Java is a set of several computer software products and specifications from Oracle Corporation that provides a system for developing application software and deploying it in a cross-platform computing environment. Java is used in a wide variety of computing platforms from embedded devices and mobile phones on the low end, to enterprise servers and supercomputers on the high end.

Python:

Python is an interpreter, interactive, object-oriented programming language. It incorporates modules, exceptions, dynamic typing, very high level dynamic data types, and classes. Python combines remarkable power with very clear syntax. It has interfaces too many system calls and libraries, as well as to various window systems, and is extensible in C or C++. It is also usable as an extension language for applications that need a programmable interface. Python is a high-level general purpose programming language that can be applied to many different classes of problems.

Raspbian:

Raspbian is a free Operating System based on Debian optimized for the raspberry pi hardware. Raspbian comes with more than 35000 packages; pre-combined software bundled in a nice format for easy installation on Raspberry pi.

Node JS:

Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient. Node.js' package ecosystem, npm, is the largest ecosystem of open source libraries in the world.

Putty: A free SSH and telnet client for windows

Putty is a free and open-source terminal emulator, serial console and network file transfer application. It supports several network protocols, including SCP, SSH, Telnet, rlogin, and raw socket connection. It can also connect to a serial port.

Putty supports many variations on the secure remote terminal, and provides user control over the SSH encryption key and protocol version, alternate ciphers such as 3DES, Arcfour, Blowfish, DES, and Public-key authentication. It also can emulate control sequences from xterm, VT102 or ECMA-48 terminal emulation, and allows local, remote, or dynamic port forwarding with SSH (including X11 forwarding). The network communication layer supports IPv6, and the SSH protocol supports the zlib@openssh.com delayed compression scheme.

It can also be used with local serial port connections.

PuTTY comes bundled with command-line SCP and SFTP clients, called "pscp" and "psftp" respectively, and plink, a command-line connection tool, used for non-interactive sessions.[7]

Conclusion

The prime objective of our project is to use the Smartphone to control the home appliances effectively. The android-app interface is used to control the home appliances. This project is based on the Raspberry pi, Android platform Java and Node JS along with python. These platforms are Free Open Source Software. So the overall implementation cost is low and can be easily configured. User can easily interact with the android phone/tablet. The user can send commands via the smartphone. The data are being analyzed by the application and are sent over a network. The Raspberry pi acts as a server, analyses the data and activates the GPIO (General Purpose Input Output) Pins. The GPIO Pins are connected to the relays switch which activated the required home appliances. In this way, automation process is carried out. This is a simple prototype. Using this as a reference further it can be expanded to many other programs.

References:

1. https://docs.python.org/
2. http://developer.android.com/training/index.html
3. http://elinux.org/RPi_Hub
4. http://www.raspberrypi.org/
5. http://stackoverflow.com/
6. http://electronics.howstuffworks.com/
7. http://cctvinstitute.co.uk/smart-home/
8. https://web.archive.org/web/20160505124414/http://www.reuters.com/article/researchand-markets-idUSnBw195490a+100+BSW20150119
9. Muhammad Izhar Ramli, Mohd Helmy Abd Wahab, Nabihah, “TOWARDS SMART

HOME: CONTROL ELECTRICAL DEVICES ONLINE”, Nornabihah Ahmad International

Conference on Science and Technology: Application in Industry and Education (2006)

Appendix A: Raspberry Pi GPIO

Introduction:

The Raspberry Pi allows peripherals and expansion boards to access the CPU by exposing the in and outputs. The production board has a 26-pin 2.54mm (100mil) expansion header, arranged in a 2x13 strip. They provide 8 GPIO pins plus access to I2C, SPI, UART), as well as +3V3, +5V and GND supply lines. Pin one is column 0 on the bottom row. Voltage levels are 3v3. There is no over-voltage protection on the board - the intention is that people interested in serious interfacing will use an external board with buffers, level conversion and analog I/O rather than soldering directly onto the main board.

It is also possible to reconfigure some of the pins to provide a second I2C interface.

Kernel boot messages go to the UART at 115200bps.

Header Pinout:

Top Row	5V0	DNC	GND	TXD	RXD	GPIO1	DNC	GPIO4	GPIO5	DNC	GPIO6	SPI_CE0_N	SPI_CE1_N
Bottom Row	3V3	SDA0	SCL0	GPIO7	DNC	GPIO0	GPIO2	GPIO3	DNC	SPI_MOSI	SPI_MISO	SPI_SCLK	DNC

Colour legend
+5V
+3.3V
Do not connect
UART
GPIO
SPI
I2C

Power Pins:

Maximum permitted current draw from the 3v3 pin is 50mA.

Maximum permitted current draw from the 5v pin is the USB input current (usually 1A) Minus any current draw from the rest of the board.

- Model A: 1000mA - 500mA -> max power draw: 500mA
- Model B: 1000mA - 700mA -> max power draw: 300mA
-

General Purpose Input /Output (GPIO):

General Purpose Input /Output (GPIO) is a generic pin on a chip whose behavior (including whether it is an input or output pin) can be controlled (programmed) through software. All the UART, SPI and I2C pins can be reconfigured as GPIO pins, to provide a total of 17 GPIO pins. Each of their functions is detailed in the Broadcom BCM2835 chipset datasheet.

The available alternative functions and their corresponding pins are detailed below. These numbers are in reference to the chipset documentation and may not match the numbers exposed in Linux or detailed above. Only fully usable functions are detailed, for some alternative functions not all the necessary pins are available for the functionality to be actually used. All exposed pins can be used for GPIO

Top Row Pinout:

Header	2	4	6	8	10	12	14	16	18	20	22	24
Chipset	14	15	18	23	24	25	8	7				
Function	5V0	DNC	GND	TXD	RXD	PWM	DNC	GPIO	GPIO	DNC	GPIO	SPI_CI

Bottom Row Pinout:

Header	1	3	5	7	9	11	13	15	17	19	21	23	25
Chipset	0	1	4	17	21	22	10	9	11				
Function	3V3	SDA0	SCL0	GPIO	DNC	GPIO	GPIO	GPIO	DNC	SPI_MOSI	SPI_MISO	SPI_SCLK	DNC

The complete list of chipset pins which are available are: 0, 1, 4, 7, 8, 9, 10, 11, 14, 15,

17, 18, 21, 22, 23, 24, 25

- Pin 12 supports PWM.

- GPIO voltage level is 3V3 and are not 5V tolerant.

- Each GPIO can interrupt, high/low/rise/fall/change.

It is also possible to reconfigure some of the pins to provide an ARM JTAG interface.

However, ARM_TMS isn't available for this (chipset pin 12 or 27 is needed). It is also possible to reconfigure some of the pins to provide an I2S (hardware mod may be required) or PCM interface.

Appendix B: Raspberry Pi Quick Start Guide

Quick Start Guide

The Raspberry Pi – Single Board Computer

Source: Raspberry Pi & Wiki

Chapter 1: RPi Hardware Basic Setup

Typical Hardware You Will Need:

While the RPi can be used without any additional hardware (except perhaps a power supply of some kind), it won't be much use as a general computer. As with any normal PC, it is likely you will need some additional hardware. The following are more or less essential:

- Raspberry Pi board
- Prepared Operating System SD Card
- USB keyboard
- Display (with HDMI, DVI, Composite or SCART input)
- Power Supply
- Cables

Highly suggested extras include:

- USB mouse
- Internet connectivity - a USB WiFi adaptor (Model A/B) or a LAN cable (Model B) Powered USB Hub

- Case

Connecting Together

You can use the diagram to connect everything together, or use the following instructions:

1. Plug the preloaded SD Card into the Pi.

2. Plug the USB keyboard and mouse into the Pi, perhaps via a USB Hub. Connect the

Hub to power, if necessary

3. Plug the video cable into the screen (TV) and into the Pi.

4. Plug your extras into the Pi (USB Wi-Fi, Ethernet cable, hard drive etc.). This is

Where you may really need a USB Hub.

5. Ensure that your USB Hub (if any) and screen are working.

6. Plug the power source into the main socket.

7. With your screen on, plug the other end of the power source into the Pi.

8. The Pi should boot up and display messages on the screen.

It is always recommended to connect the MicroUSB Power to the unit last (while Most connections can be made live; it is best practice to connect items such as displays/h/w pin connections with the power turned off).

The RPi may take a long time to boot when powered-on for the first time, so be patient!

Prepared Operating System SD Card:

As the RPi has no internal storage or built-in operating system it requires an SD-Card that Is set up to boot the RPi.

1) You can create your own preloaded card using any suitable SD card you have. Be

Sure to backup any existing data on the card.

2) Preloaded SD cards will be available from the RPi Shop.

This guide will assume you have a preloaded SD card.

Keyboard & Mouse

Most standard USB keyboards and mice will work with the RPi. Wireless Keyboard /mouse should also function, and only require a single USB port for an RF dongle. In order to use a Bluetooth keyboard or mouse you would need to use a Bluetooth dongle, which again uses a single port.

Remember that the Model A has a single USB port and the Model B only has two (typically a keyboard and mouse will use a USB port each).

Display:

There are two main connection options for the RPi display, HDMI (high definition) and Composite (low definition). HD TVs and most LCD Monitors can be connected using a full-size 'male' HDMI cable, and with an inexpensive adaptor if DVI is used. HDMI versions 1.3 and 1.4 are supported, and a version 1.4 cable is recommended. The RPi outputs audio and video via HMDI, but does not support HDMI input. Older TVs can be connected using Composite (a yellow-to-yellow cable) or via SCART (using a Composite to SCART adaptor). PAL and NTSC TVs are supported. When using composite video, audio is available from a 3.5mm (1/8 inch) socket, and can be sent to your TV, to headphones, or to an amplifier. To send audio your TV, you will need a cable which adapts from 3.5mm to double (red and white) RCA connectors.

Note: There is no VGA output available, so older VGA monitors will require an expensive adaptor.

Using an HDMI to DVI-D (digital) adaptor plus a DVI to VGA adaptor will not work.

HDMI does not supply the DVI-A (analogue) needed to convert to VGA - converting an HDMI or

DVI-D source to VGA (or component) needs an active converter. (It can work out cheaper to buy a new monitor.) The lack of VGA has been acknowledged as a priority issue.

Power Supply:

The unit uses a Micro USB connection to power itself (only the power pins are connected – so it will not transfer data over this connection). A standard modern phone charger with a micro-USB connector will do, but needs to produce at least 700mA at 5 volts. Check your power supply's ratings carefully. Suitable mains adaptors will be available from the RPi Shop and are recommended if you are unsure what to use.

You can use a range of other power sources (assuming they are able to provide enough current ~700mA):

- Computer USB Port or powered USB hub (will depend on power output)
- Special wall warts with USB ports
- Mobile Phone Backup Battery (will depend on power output) (in theory – needs confirmation)

To use the above, you'll need a USB A 'male' to USB micro 'male' cable - these are often shipped as data cables with MP3 players.

Cables:

You will probably need a number of cables in order to connect your RPi up.

1. Micro-B USB Power Cable

2. HDMI-A or Composite cable, plus DVI adaptor or SCART adaptor if required, to connect your RPi to the Display/Monitor/TV of your choice.

3. Audio cable, this is not needed if you use a HDMI TV/monitor.

4. Ethernet/LAN Cable

Additional Peripherals:

You may decide you want to use various other devices with your RPi, such as Flash Drives/Portable Hard Drives, Speakers etc.

Internet Connectivity:

This may be an Ethernet/LAN cable (standard RJ45 connector) or a USB Wi-Fi adaptor. The RPi Ethernet port is auto-sensing which means that it may be connected to a router or directly to another computer (without the need for a crossover cable).

USB-Hub:

In order to connect additional devices to the RPi, you may want to obtain a USB Hub, which will allow multiple devices to be used. It is recommended that a powered hub is used this will provide any additional power to the devices without affecting the RPi itself.

USB version 2.0 is recommended. USB version 1.1 is fine for keyboards and mice, but may not be fast enough for other accessories.

Case:

Since the RPi is supplied without a case, it will be important to ensure that you do not use it in places where it will come into contact with conductive metal or liquids, unless suitably protected.

Expansion & Low Level Peripherals:

If you plan on making use of the low level interfaces available on the RPi, then ensure you have suitable header pins for the GPIO (and if required JTAG) suitable for your needs. Also if you have a particular low-level project in mind, then ensure you design in suitable protection circuits to keep your RPi safe.