Deepak Soni | February 20, 2015

Objective:

Alfred

Wi-Fi roter cum Network attached storage cum home automation

To make a Wi-Fi router out of the Raspberry Pi to access internet and also use this network to control and automate your house.

Prerequisites:

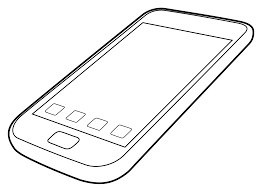
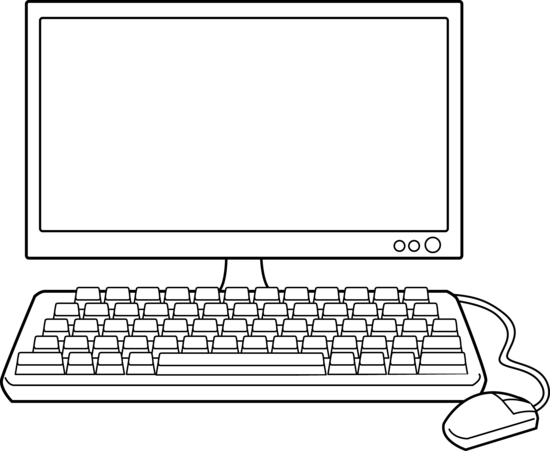
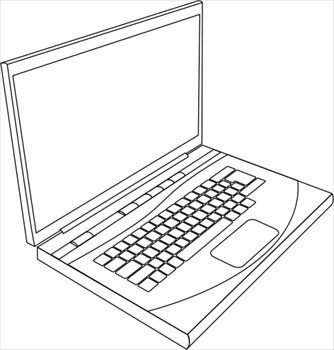
* Basics of C and php
* Command line environment in Linux
* Basics of UART communication.
* Arduino programming
* Networking and Apache server

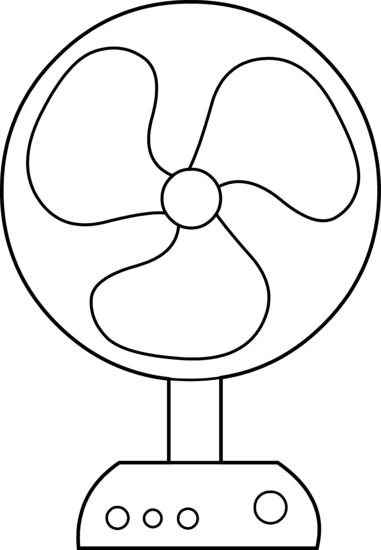
Hardware Requirements:

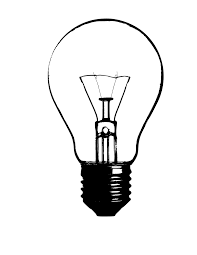
* Raspberry pi (model B)
* A 4 GB SD card
* Arduino Mega 2560
* USB Wi-Fi adapter
* 5V power supply
* ULN2003 IC

Approach:

In this project, we are making Raspberry pi to work as a Wi-Fi hotspot, which will create a local wireless network and as the pi (Raspberry pi) is connected to the internet anyone connected to this network can access internet and also control the pi. The pi is also connected to an Arduino through UATR port, and as an Arduino can control any home equipment through relay, we can control any home equipment through pi wirelessly.







Procedure:

* First we have to set up Raspberry pi SSH and remote access it for this you can follow any tutorial on internet.
* Once accessible through LAN cable, we proceed to make it wireless, Insert the Wi-Fi adapter through USB and follow the procedure given on [www.elinux.org/RPI-Wireless-Hotspot](http://www.elinux.org/RPI-Wireless-Hotspot). Now your raspberry pi will be creating a Wi-Fi network with internet accessibility. With this we have created a Wi-Fi router cum Network attached storage, to increase the storage capacity you can add an external USB hard disk to it.
* For the home automation part, we have to install apache and php in pi. For this follow any tutorial on how to install lamp server in raspberry pi.
* Once lamp is installed, we need to write a C code to send data to Arduino through UART port and a php script to execute the C code by calling a webpage.
* On the Arduino side we have to connect a relay through the IC ULN2003 as Arduino cannot drive a relay, also connect pi to Arduino through UART. The code in Arduino is simply an if-else statement on what data has been received from pi, for a specific data corresponding relays is turned on off. The UART port of pi is disable in factory setting so you have to turn it on for once.

Key code snippets:

The function to send null terminated string to Arduino

//From C code

void myWrite(unsigned char tmp2[])

{

int i = 0;

p\_tx\_buffer = &tx\_buffer[0];

while(tmp2[i] != '\0')

{

\*p\_tx\_buffer++ = tmp2[i];

i++;

}

\*p\_tx\_buffer++ = '\0';

if (uart0\_filestream != -1)

{

int count = write(uart0\_filestream, &tx\_buffer[0], (p\_tx\_buffer - &tx\_buffer[0])); //Filestream, bytes to write, number of bytes to write

//printf("\nData sent %s", tmp2);

if (count < 0)

{

printf("UART TX error\n");

}

}

}

void myRead()

The function to read null terminated string from Arduino

{

int flag = 0;

co = 0;

if (uart0\_filestream != -1)

{

while(flag == 0)

{

unsigned char rx\_buffer[256];

int rx\_length = read(uart0\_filestream, (void\*)rx\_buffer, 255); //Filestream, buffer to store in, number of bytes to read (max)

if (rx\_length < 0)

{

}

else if (rx\_length == 0)

{

}

else

{

int k = 0;

for(k; k < rx\_length; k++)

{

dat[co] = rx\_buffer[k];

co++;

if(rx\_buffer[k] == '\0')

{

flag = 1;

//printf("\nData recieved = %s",dat);

break;

}

}

}

}

}

Main program, initializes UART and sends a character to arduino

}

int main()

{

printf("Programme Started");

uart0\_filestream = open("/dev/ttyAMA0", O\_RDWR | O\_NOCTTY | O\_NDELAY); //Open in non blocking read/write mode

if (uart0\_filestream == -1)

{

//ERROR - CAN'T OPEN SERIAL PORT

printf("Error - Unable to open UART. Ensure it is not in use by another application\n");

}

Setting UART parameteres

struct termios options;

tcgetattr(uart0\_filestream, &options);

options.c\_cflag = B19200 | CS8 | CLOCAL | CREAD;

options.c\_iflag = IGNPAR;

options.c\_oflag = 0;

options.c\_lflag = 0;

tcflush(uart0\_filestream, TCIFLUSH);

tcsetattr(uart0\_filestream, TCSANOW, &options);

printf("\nRPI UART Initialised");

Sends ‘’1’’ to Arduino to indicate turning on the light

unsigned char give[] = "1";

myWrite(give);

myRead();

}

Reads acknowledge from Arduino

//From Arduino.ino

void setup()

{

Serial.begin(9600);

Initializing serial port in Arduino

pinMode(13,OUTPUT);

}

void loop()

Pin 13 is connected to relay uln IC

{

if(Serial.available)

{

char ch = Serial.read();

Turns on 13 if ‘1’ is received

if(ch == '1')

digitalWrite(13,HIGH);

else

if(ch == '0')

Turns off 13 if ‘0’ is received

digitalWrite(13,LOW);

}

}

Applications:

* Home control and automation through local network or internet
* Connectivity with multiple devices
* Wi-Fi router
* Personal cloud network
* Network attached storage
* Can be used as local low power limited internet access for areas with poor internet availability

Links:

<https://github.com/Deepak61900/Alfred>

deepak61900@gmail.com

The Verdict:

This project is essentially useful for the projects of Home automations and local networking. The product can be used for sharing a stored data in a local area. Realizing the importance of internet for everyone, this product is the solution to create limited accessibility for everyone, even when they are not connected to the Internet. Currently maybe for institutions like schools of rural areas where no internet can reach it can provide a local file sharing spot on which things like books or ever the whole Wikipedia can be can be accessed. The solution is very low cost and low power and is very suitable for countries like India.