23/09/2019

Parts needed for Project

Raspberry Pi 4, 4Gb ram €49.63

RS components -> stock No. 182-2096

Auvidea B101 HDMI to CSI-2 bridge

€69.90 Auvidea website

Raspberry pi 4 Plug €12.17

Farnell -> 3106944

Raspberry Pi, relay board shield €30.78

RS components -> Stock No. 174-3270

or

✓ Farnell -> 2434230 €29.00

or

Farnell -> 2451886 €14.25

Git username: Noddy26

24/09/2019

Name for Project

remote Gaming Server

Game anywhere Server

Gaming world-web emulator

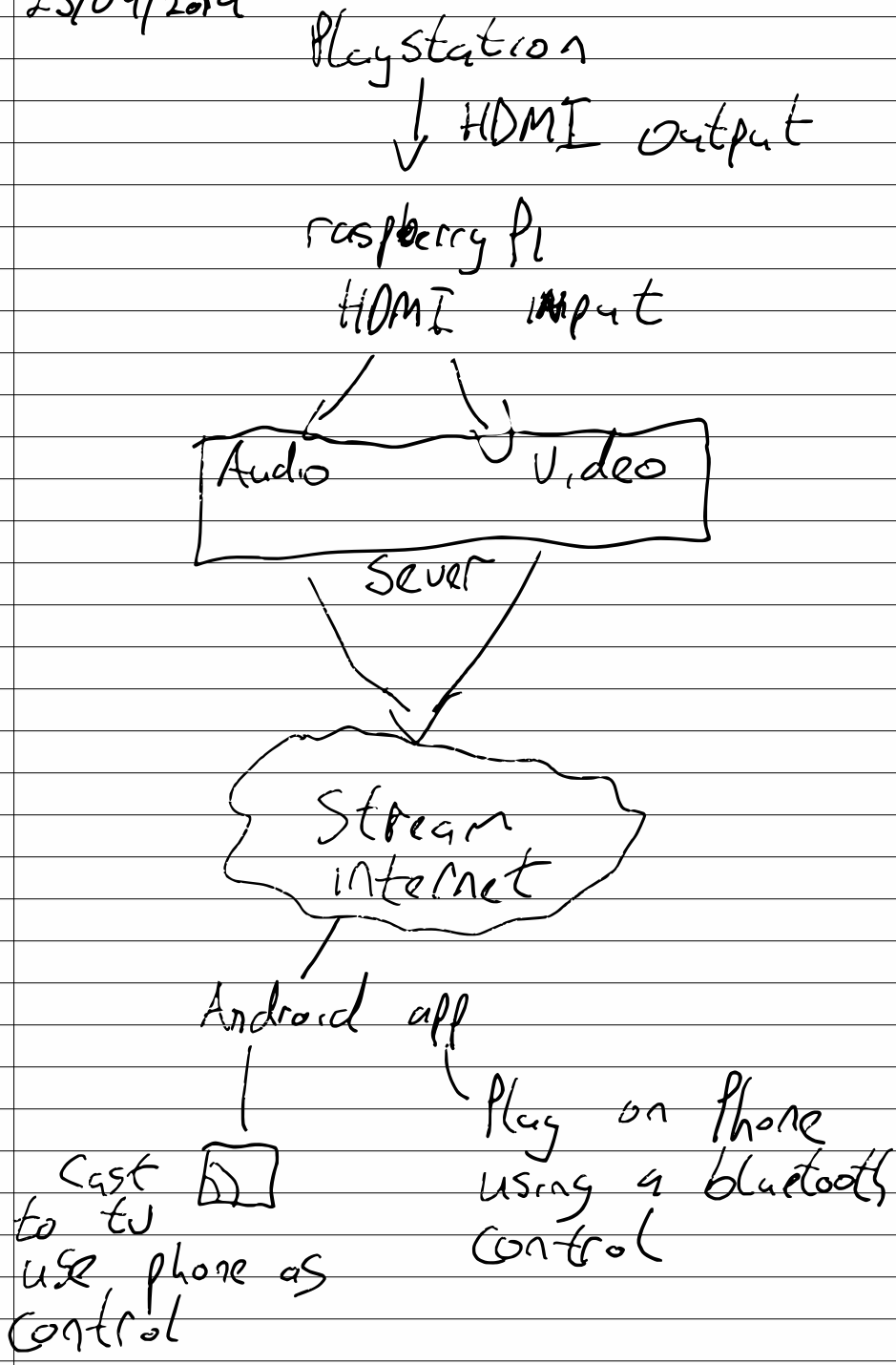
Playstation streamer

Phone Gamer

Playstation App player

Playstation App Emulator

25/09/2019

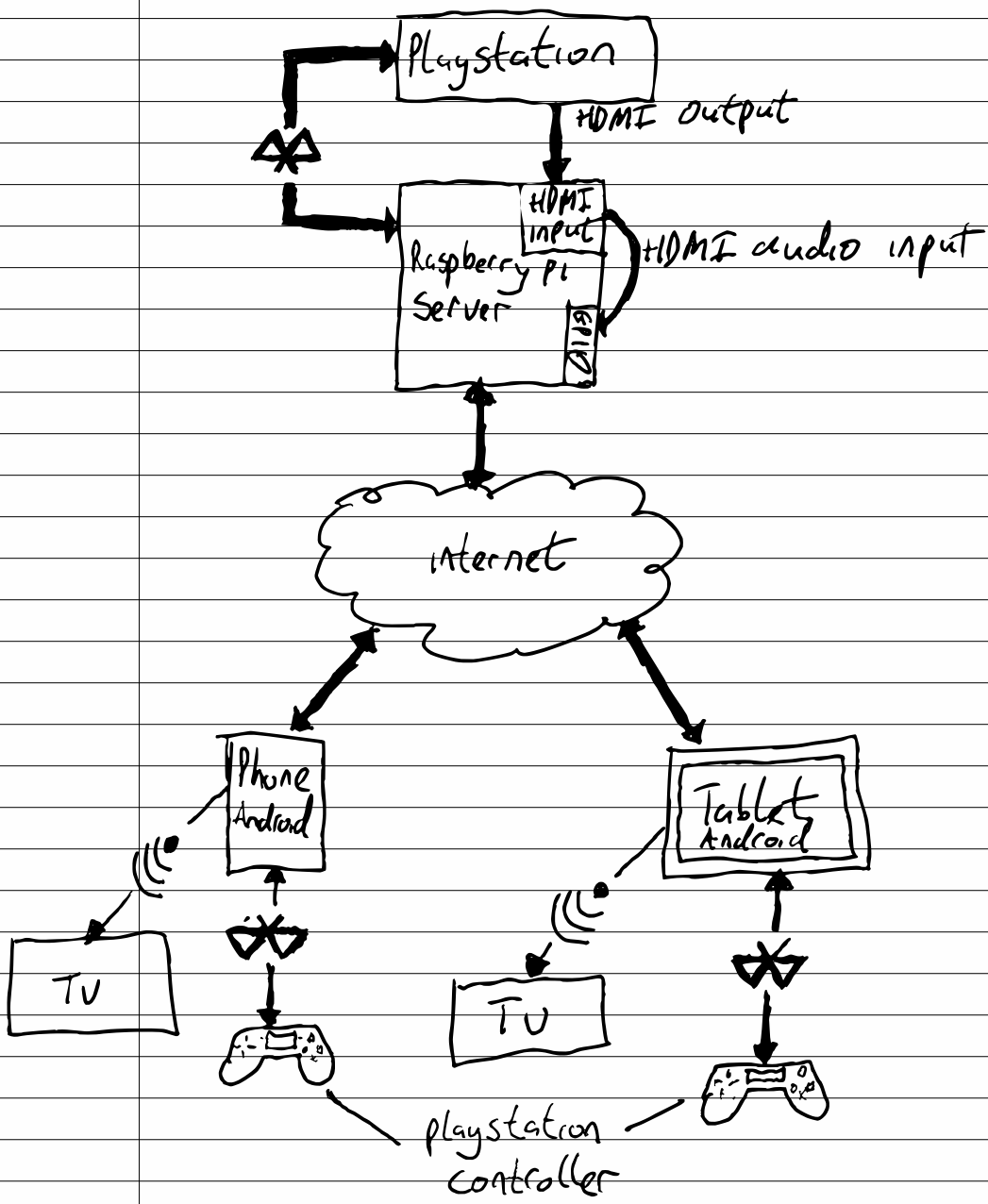

Playstation
HDMI output
raspberry Pi
HDMI input
Audio
Video
Sever
Stream
internet
Android app
Cast
to tv
use phone as
control
Play on Phone
using a bluetooth
control

1. Get HDMI input working
2. Look into digital Audio input GPIO Pins
3. Server side Pie, Python script to break video into packets and Audio send it over internet

Audio and video must be synced before being streamed

Android app will make a request to the server pi to stream video and audio to the phone. The app will have two choices. First choice will be to play video on the phone and connect a playstation controller to the phone and send button presses back to the server to send them to the Playstation. Second choice will be to cast video and audio to a tv and either use the phone as a controller or connect Playstation controller to the phone, both choices will send button presses back to the server to communicate with the Playstation.

30/09/2019


Playstation
HDMI output
HDMI input
Raspberry Pi Server
USB
HDMI audio input
Internet
Phone Android
Tablet Android
TV
TV
playstation controller

1/10/2019

Raspberry Pi GPIO Pins

Power	3v3	1	2	5v	Power
SDA1_I2C	GPIO 2	3	4	5v	Power
SCL1_I2C	GPIO 3	5	6	Gnd	
	GPIO 4	7	8	GPIO 14	UART_TxD
	GND	9	10	GPIO 15	UART_RxD
	GPIO 17	11	12	GPIO 18	PCM_CLK
	GPIO 27	13	14	GND	
	GPIO 22	15	16	GPIO 23	
Power	3v3	17	18	GPIO 24	
SPI0_MOSI	GPIO 10	19	20	GND	
SPI0_MISO	GPIO 9	21	22	GPIO 25	
SPI0_SCLK	GPIO 11	23	24	GPIO 8	SPI0_CE0_N
	GND	25	26	GPIO 7	SPI0_CE1_N
I2C_ID_EEPROM	ID_SD	27	28	ID_SC	I2C_ID_EEPROM
	GPIO 5	29	30	GND	
	GPIO 6	31	32	GPIO 12	
	GPIO 13	33	34	GND	
	GPIO 19	35	36	GPIO 16	
	GPIO 26	37	38	GPIO 20	
	GND	39	40	GPIO 21	

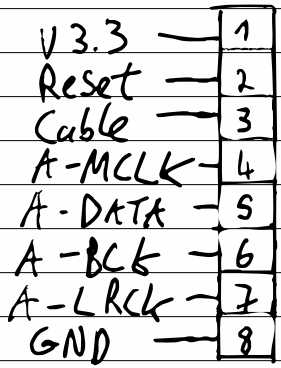
Server will be Programmed in python on Raspberry Pi, and some bash scripts.

The client android app will be programmed in Java

1/10/2019

Auvidea chip Audio output

	Vpp	Vmax	Period	freq



HDMI-in 2.60V HDMI-out 1.16V

6.56V, 4.96V, 8LDOs, 12.3Mhz

Pin	Function	Description
1	V3.3	V3.3 Power out (on Board LDO with 300mA max)
2	Reset	hardware reset (3.3V level, low active)
3	Cable	Cable Detect: high (3.3V) HDMI cable Plugged-in
4	A-MCLK	I2S audio master clock (3.3V level)
5	A-DATA	I2S audio data out (3.3V level)
6	A-BCK	I2S audio bit clock (3.3V level)
7	A-LRCK	I2S audio word clock (3.3V level)
8	GND	Ground (0V)

12	GPIO 18	BCK	Check all
35	GPIO 19	LRCK	Signals on
37	GPIO 20	DATA-IN	a
40	GPIO 21	DATA-Out	

Pin 5 → GPIO 20
Pin 6 → GPIO 18
Pin 7 → GPIO 19

reset - GPIO 24
Cable - GPIO 23

4/10/19

Installing openCV

you have to delete CMakeCache.txt before running Cmake or else you will get errors and won't be able build.

https://docs.opencv.org/master/d7/d9f/tutorial_linux_install.html

Link with full dependencies to install for openCV

Installing vim text editor

Sudo apt-get install vim

Installing mySQL

Sudo apt-get update
Sudo apt-get install mysql-server

Allow remote access

Sudo ufw enable
Sudo ufw allow mysql

Start the server

Sudo systemctl start mysql

Full command ↳

Launch mysql as root user

/usr/bin/mysql -u root -p

4/10/19

check all packages in an import

for each in dars("module Name"): Print each

Python 2.7 for pyaudio

I2c clock bps (bits per second)

125 Mb/s and 3826 b/s

Speeds higher than 30 Mb/s won't work

Sudo modprobe i2c_bcm2708 baudrate=

set the speed

HDMI

uncompressed Audio

8 channels into cable

HDMI DATA Tranfer Rate

Camera serial interface

mp3 128kps

Setting up the server 4/10/19

Install apache2 Gaming db

Create a Database

install mysql

Sudo apt install mariadb-server

Sudo mysql_secure_installation

Sudo mysql -u root -p

CREATE DATABASE "Name";

CREATE USER "user" '@' 'localhost' IDENTIFIED BY 'password';

GRANT ALL PRIVILEGES ON exampledb.* To 'Gaming db' '@' 'localhost';

FLUSH PRIVILEGES

USE "Name_of_database" New users

CREATE TABLE "name"

HDMI Interface 6/10/19

High-definition Multimedia

Pin 1 TMDS Data2+
Pin 2 TMDS Data2 Shield
Pin 3 TMDS Data2-
Pin 4 TMDS Data1+
Pin 5 TMDS Data Shield
Pin 6 TMDS Data1-
Pin 7 TMDS Data0+
Pin 8 TMDS Data0 Shield
Pin 9 TMDS Data0-
Pin 10 TMDS Clock+
Pin 11 TMDS Clock Shield
Pin 12 TMDS Clock-
Pin 13 CEC
Pin 14
Pin 15 SCL (I^2C Serial clock For DDC)
Pin 16 SDA (I^2C Serial data For DDC)
Pin 17 Ground (For DDC, CEC, ARC, and HEC)
Pin 18 +5V (Min. 0.055 A)
Pin 19 Hot Plug detect

Bitrate up to 48Gbps

uncompressed video and Audio

No Signal conversion necessary, No loss of Quality

digital Audio
8 channels
16-bit 20-bit 24-bit
Sample rates
32kHz - 192kHz
44,100 samples Per second
16 bit

CEC ↓ Consumer Electronics Control

HEC ↓ HDMI Ethernet channel

Hot plug ↓ Check if Connection is Live

TMDS - Transition-minimized differential signaling
For transmitting high speed serial data

DDC - Display DATA Channel - check what audio video formats it Supports

6/10/19

HDMI Bit Depth – Color Sampling

8-Bit

256 Shades

red 256 × Green 256 × Blue 256

16 million

10-Bit

1024 Shades

red 1024 × Green 1024 × Blue 1024

1 Billion

High bit Depth more colors can be captured

4:4:4 – full color

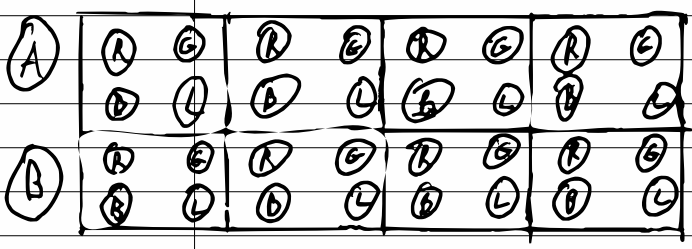
4:2:2 – 1/2 Color

4:2:0 – 1/4 Color

Luma is Relative Luminance of a weighted sum of Linear RGB components

Luma | color Row A | color Row B

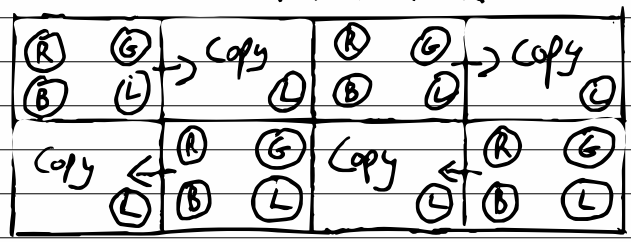
4:4:4



Row B will sample the opposite of Row A to spread the image evenly

Luma | color Row A | color Row B

4:2:2



Every Luma channel is sampled. Only samples 2 of the 4 pixles The other pixles are copied which will reduce file size

10/10/19

Auvidea B101 HDMI to CSI-2 Bridge

HDMI signals does not match the signals expected into the camera Serial Interface.

TPD12S521

electro-static discharge circuit protection device for high-definition multimedia interface.

High speed transmission line with level translation and Buffering for HDMI DDC, CEC and hot plug signals

Is used for a HDMI receiver

NXP Design
http://www.nxp.com/pip/1P4786CZ32.html

B101 HDMI to CSI-2 bridge

The bridge is needed to convert the HDMI input data to the camera Serial interface required data so it can processed by the SOC.

11/10/19

Why Serial transmission is used?

Serial communication uses fewer wires, this makes serial communicaton signals clearer. The signal frequency can be increased, Parallel need to arrive at the same time.

Serial transmissio is slower than Parallel when the signal frequency is the same.

Parallel can transfer one word per cycle 8bits Serial 1 bit

Parallel signal frequency has a Limit all signals must arrive together with high frequency's this can not be guaranteed.

Waiting for each bit to sync up will lower transfer rate

The higher signals can result in crosstalk with Parallel signals and the word can get corrupted.

Serial transfers less data per cycle but can go to much high frequencies, which result in a higher net transfer rate

Synchronization is also a big Problem when Parallel signal frequency gets higher.

13/10/19

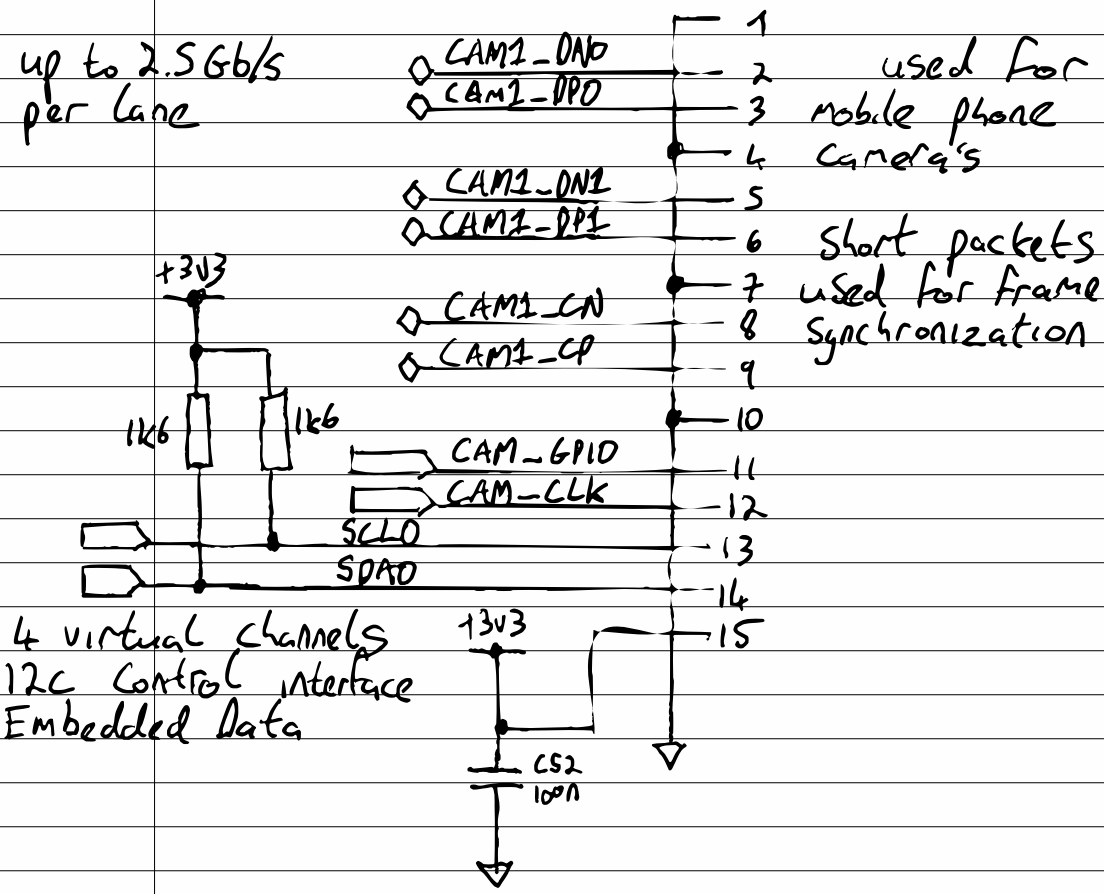
Raspberry Pi Camera Serial Interface

15 pin Socket

Allows data rates up to 800mbps per serial pin

1Gbps each pin set as the limit

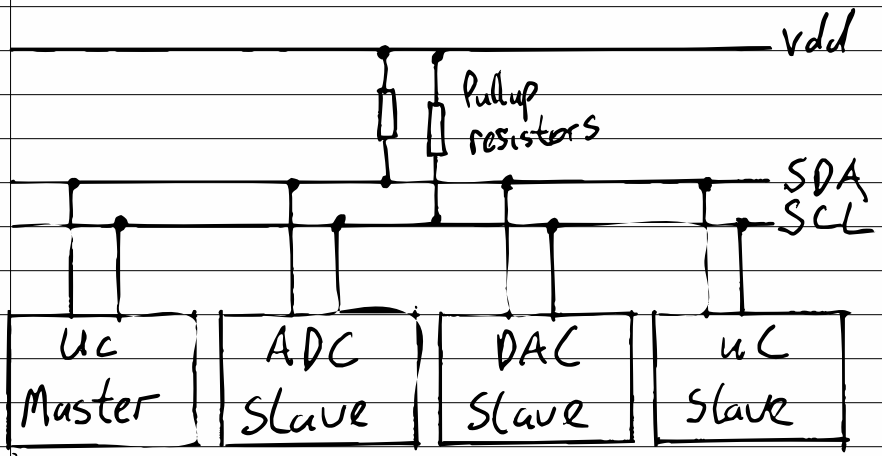
CSI-2 directly to SoC processor



14/10/19

Raspberry Pi I2c for Audio input

I2c is Serial Protocol for two-wire interface to connect low-speed devices like microcontrollers, EEPROMs, A/D and D/A converters, I/O interfaces and other similar peripherals in embedded systems. Each I2C slave needs an addresss.



up to 400kHz Transfers 8 bits or bytes

Each slave device has 7-bit Address.
other bit is used to signal reading from or writing to the device.

Master device generates the clock, doesn't need an address.

Can have up to 128 devices connected
Raspberry Pi has 117 available address

24/10/2019

Making a Static Ip Address

Raspian by default uses dhcpcd to manage network interfaces.

vim /etc/dhcpcd.conf

Change wifi

Sudo iwlist wlan0 scan

Sudo Vim /etc/wpa_supplicant/wpa_supplicant.conf

check change = ifconfig wlan0

Sudo reboot

Ip Address 192.168.0.102

Sudo vim /etc/resolv.conf

Nameserver 192.168.0.1

Sudo vim /etc/dhcpcd.conf

Interface wlan0
Static ip_address = 192.168.0.13/24
Static routers = 192.168.0.1
Static domain_name_servers = 192.168.0.1 8.8.8.8

25/10/2019

Commands for monitoring

CPU Temp - /opt/vc/bin/vcgencmd measure_temp

Top -b | head -n 5

↳ Give memory usage
CPU usage
Tasks running

127.0.0.53

192.168.0.103 192.168.0.255

28/10/2019

Python Flask Server app is For Registering an account and getting confirmation of registration, then being able to download a mixture of a Java GUI and Python GUI, which also consists of a login, this login GUI connects to a Java Server on the pi which will verify a user from the mySQL Database on the PI, then the main GUI written in python will appear and a second server will be started to stream video and audio to the python GUI, so the user can interface with the input from the playstation.

Connections will be added for playstation controller to be connected to the GUI via bluetooth and the signals from the controller will be sent back to the server and relayed to the playstation, so the user can play the game remotely.

1/11/2019

Installing Python 3.7.2 on Raspian

```
Sudo apt-get update -y
Sudo apt-get install build-essential tk-dev libncurses5-dev libncursesw5-dev
libreadline6-dev libdb5.3-dev libgdbm-dev libsqlite3-dev libssl-dev libbz2-dev
libexpat1-dev liblzma-dev zlib1g-dev libffi-dev -y
```

Install build Tools required

get python package

```
Wget http://www.python.org/ftp/python/3.7.2/Python-3.7.2.tar.xz
```

Then un-tar the file

```
tar xf Python-3.7.2.tar.xz
cd Python-3.7.2
./configure
make -j 4
Sudo make altinstall
```

02/11/2019

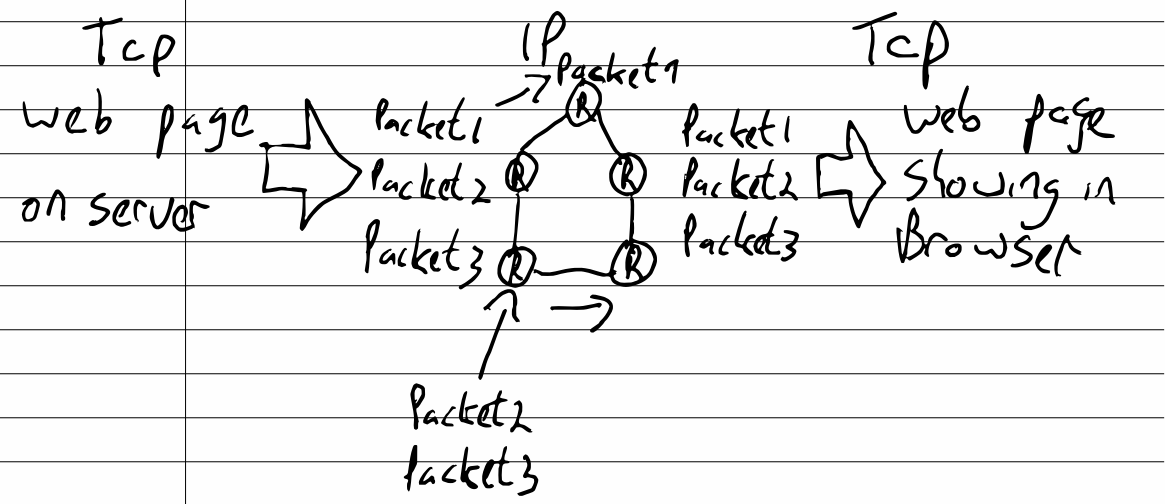
Tcp Broadcasting

Transmission Control Protocol is a protcol that establishes and maintains a network conversation, were client and server can exchange data.

Layers of Tcp

The Link layer is the physical network equipment used to interconnect nodes and servers
The internet layer connects hosts to one another across networks
The transport layer resolves all host-to-host communication
The application layer is utilzed to ensure communication between applications on a network

Tcp is reliable, it provides the recovery of segments that get lost, damaged, duplicate or received packets out of order. Tcp also requires that an acknowledgement message is received.



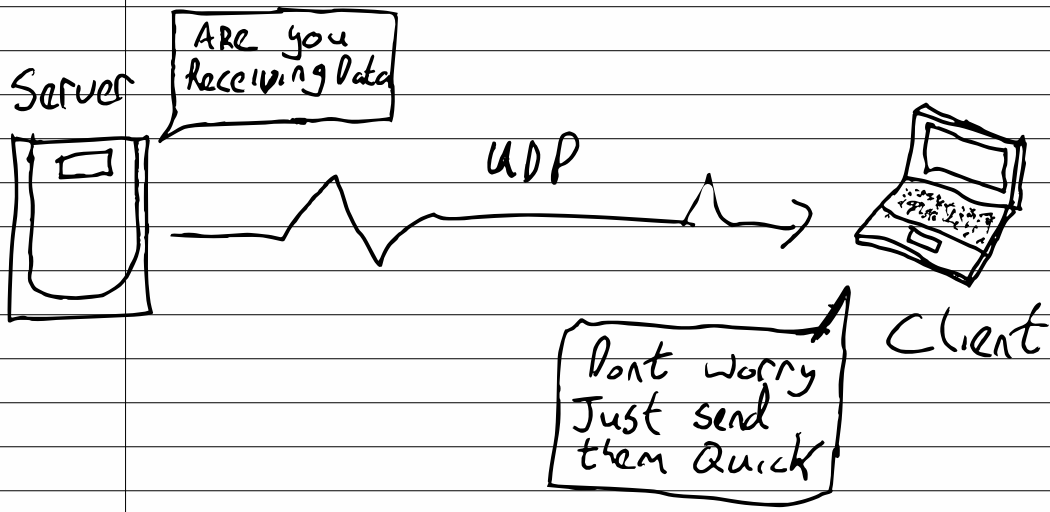
05/11/2019

UDP Broadcasting

User Data Protcol works by encapsulating data in a UDP packet The data consists of source and destination Ports to communicate on, the packet length and check-sum. When the packet is encapsulated they are sent to the client.

UDP is connectionless internet protocol. Sends out files doesnt check if they arrived.

error recovery is not Attempted
UDP header size is 8 bytes



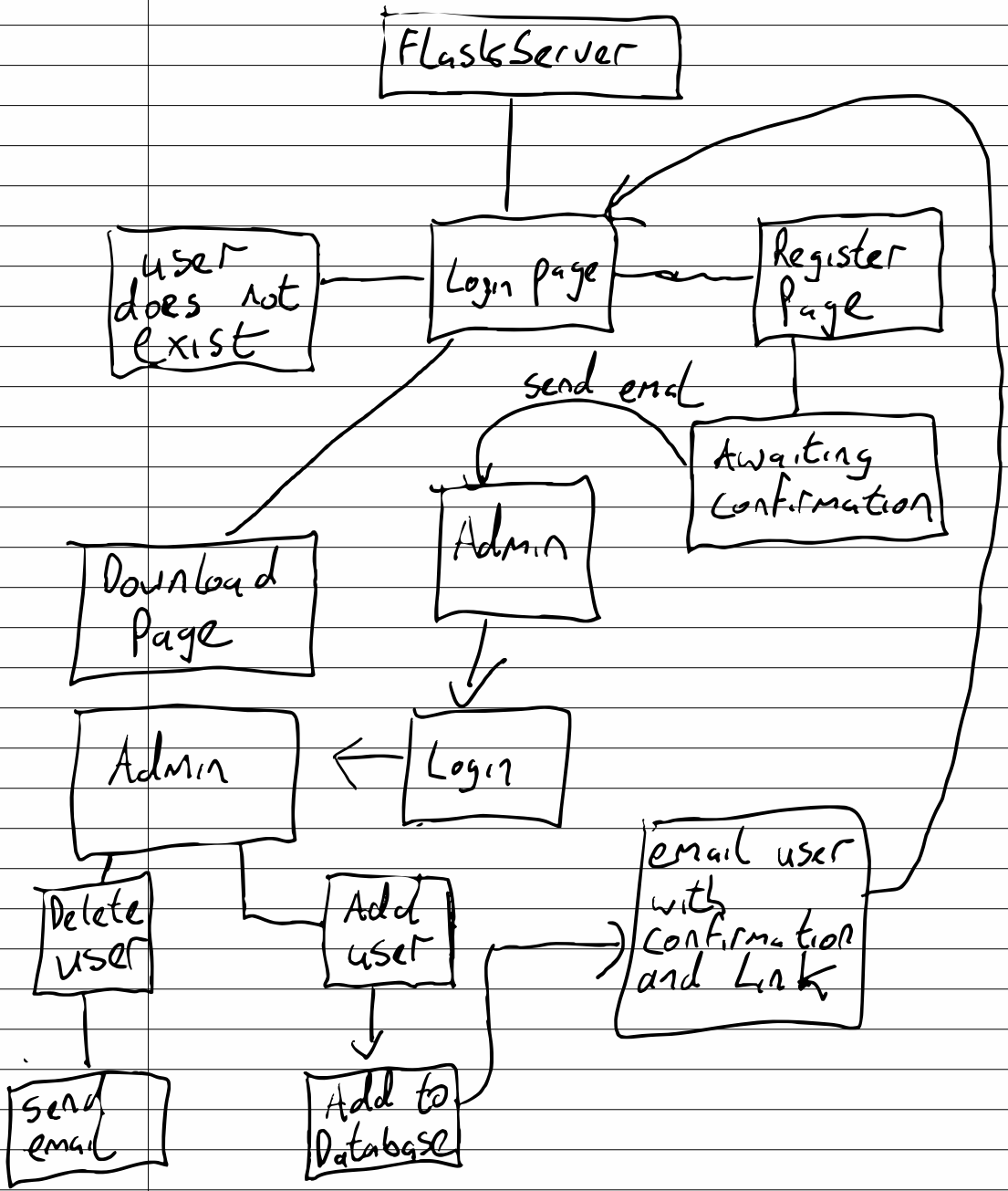
05/11/2019

Transmission Control Protocol	User DataGram Protocol
Tcp is connection-oriented	UDP is the datagram-orientated
Tcp is reliable it guarantees delivery of data to destination	UDP delivery of data to the destination can not be guaranteed
Tcp provides extensive error checking, it provides flow control and acknowledgment of data.	UDP only has basic error checking - checksum
Squencing of data is a feature of tcp(Packets arrive in order)	Data is not squenced this has to be done from Application Layer
Tcp is comparatively slower than UDP	UDP is faster, simpler and more efficient than tcp
Tcp is heavy weight	UDP is Light weight
Tcp doesn't support broadcasting	UDP supports broadcasting
Retransmission of lost Packets is possible in tcp	There is no retransmission of lost-Packets

I am going to use Tcp for streaming the video and audio from my raspberry pi server, it may be slower than UDP but reliability is what I am looking for

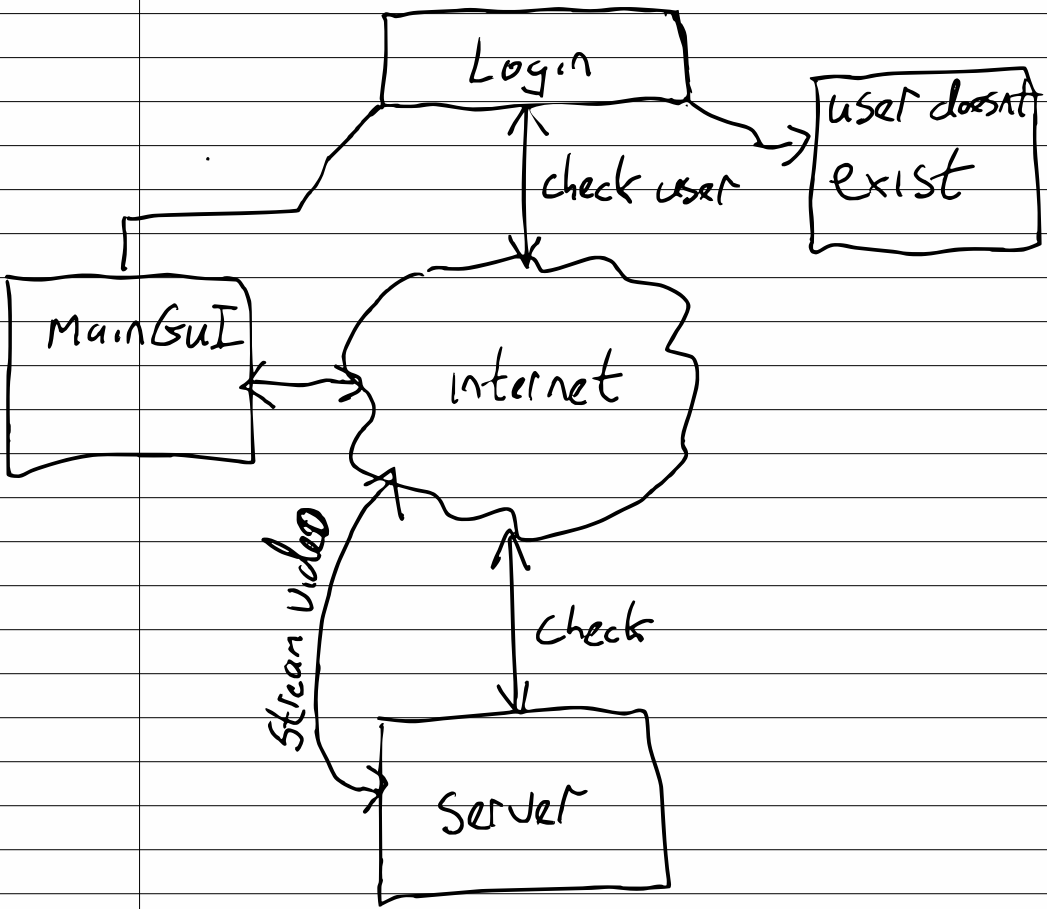
12/11/2019

Registration Server



12/11/2019

JAVA Stream Server and JAVA GuI Client



12/11/2019

Base64 encoding

Base64 encoding is the process of converting binary into limited character set of 64 characters. The characters are [A-Z], [a, z], [0-9], [+] and [/]. They all make up 64 characters.

The base64 encoded data ends up being longer than the original data, because for every 3 bytes of binary data, there are at least 4 bytes of Base64 encoded data. This is due to squeezing the data into smaller sets of characters.

Base64 Steps

- The text is converted into its respective decimal value (ASCII value).
- The decimal values are then converted into their binary equivalents.
- The binary values are concatenated into one
- The binary concatenation is divided into sections with each section containing only 6 bits.
- The equal sets of 6 bits are converted into their decimal equivalents.
- Finally, the decimal values are converted into their base64 values.