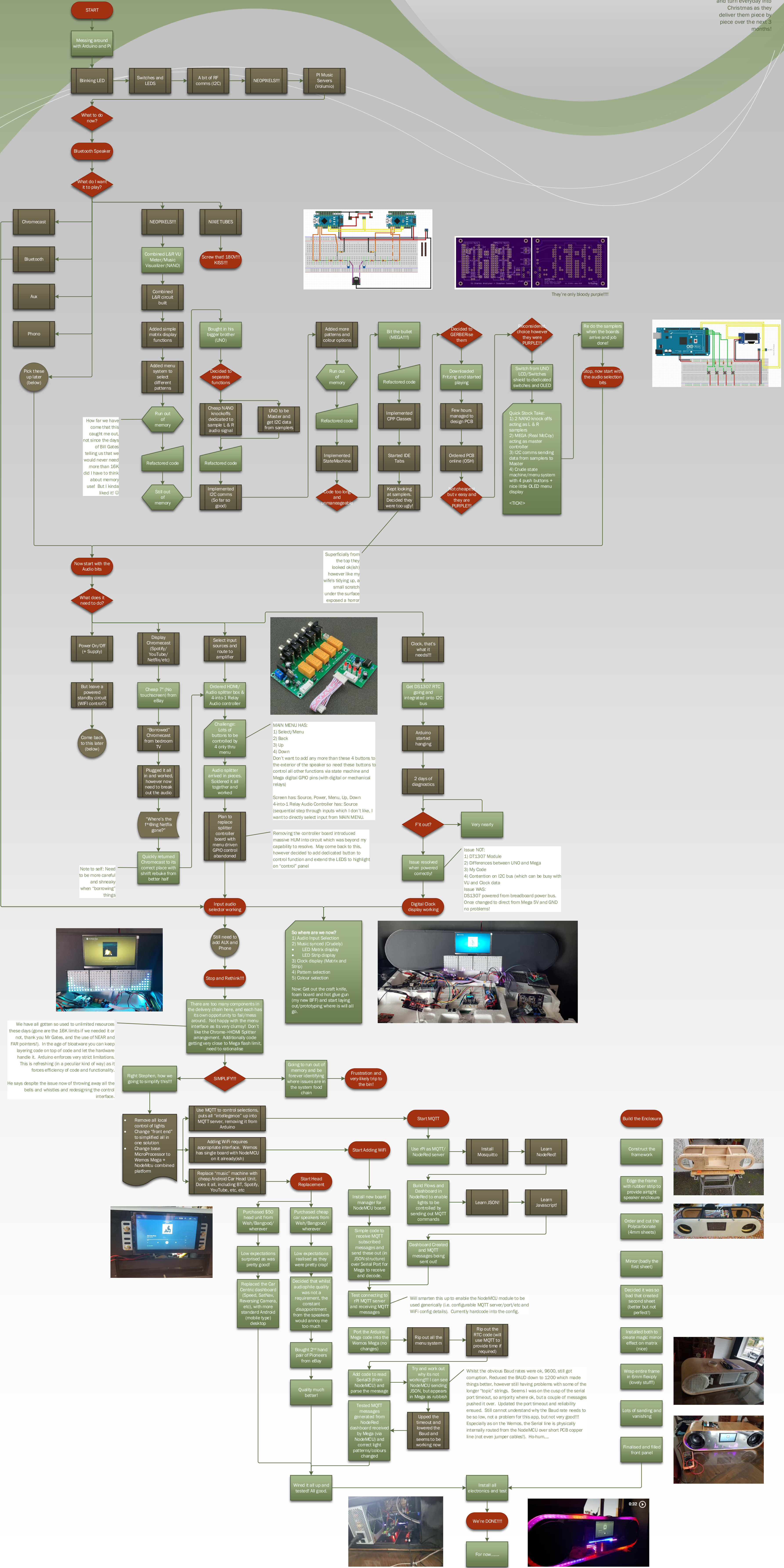


Top-Tip  
Order lots of parts from WISH in one go and turn everyday into Christmas as they deliver them piece by piece over the next 3 months!



START  
Messing around with Arduino and Pi

Blinking LED  
Switches and LEDs  
A bit of RF comms (I2C)  
NEOPixels!!!  
Pi Music Servers (Volumio)

What to do now?  
Bluetooth Speaker  
What do I want it to play?

Chromecast  
Bluetooth  
Aux  
Phono

NEOPixels!!!  
NIXIE TUBES  
Combined L&R VU Meter/Music Visualizer (NANO)  
Screw that! 180V!!! KISS!!!  
Combined L&R circuit built  
Added simple matrix display functions  
Added menu system to select different patterns  
Run out of memory  
Refactored code  
Still out of memory

How far we have come that this caught me out, not since the days of Bill Gates, telling us that we would never need more than 16K did I have to think about memory use! But I kinda liked it!

Now start with the Audio bits  
What does it need to do?

Power On/Off (+ Supply)  
But leave a powered standby circuit (WiFi control?)  
Come back to this later (below)

Display (Chromecast (Spotify, YouTube/Netflix/etc))  
Cheap 7" (No touchscreen) from eBay  
"Borrowed" Chromecast from bedroom TV  
Plugged it all in and worked, however now need to break out the audio  
"Where's the f\*cking Netflix gone?"  
Quickly returned Chromecast to its correct place with shift rebate from better half

Select input sources and route to amplifier  
Ordered HDMI/Audio splitter box & 4-in-to-1 Relay Audio controller  
Challenge: Lots of buttons to be controlled by 4 only thru menu  
Audio splitter arrived in pieces. Soldered it all together and worked  
Plan to replace splitter controller board with menu driven GPIO control abandoned

MAIN MENU HAS:  
1) Select/Menu  
2) Back  
3) Up  
4) Down  
Don't want to add any more than these 4 buttons to the exterior of the speaker so need these buttons to control all other functions via state machine and Mega digital GPIO pins (with digital or mechanical relays)

Screen has: Source, Power, Menu, Up, Down  
4-in-to-1 Relay Audio Controller has: Source (sequential step through inputs which I don't like, I want to directly select input from MAIN MENU)

Removing the controller board introduced massive HUM into circuit which was beyond my capability to resolve. May come back to this, however decided to add dedicated button control function and extend the LEDs to highlight on "control" panel

Note to self: Need to be more careful and sneaky when "borrowing" things

Input audio selector working  
Digital Clock display working

So where are we now?  
1) Audio Input Selection  
2) Music/sync'd (Crude!)  
• LED Matrix display  
• LED Strip display  
3) Clock display (Matrix and Strip)  
4) Pattern selection  
5) Colour selection  
Now: Get out the craft knife, foam board and hot glue gun (my new BFF) and start laying out/prototyping where it will all go.

There are too many components in the delivery chain here, and each has its own opportunity to fall/mess around. Not happy with the menu interface as it's very clumsy. Don't like the Chrome-HDMI Splitter arrangement. Additionally code getting very close to Mega flash limit, need to rationalise

Right Stephen, how we going to simplify this!!!  
SIMPLeFY!!!  
Going to run out of memory and be forever identifying where issues are in the system food chain  
Frustration and very likely trip to the bin!

Remove all local control of lights  
Change "front end" to simplified all in one solution  
Change base MicroProcessor to Wemos Mega + NodeMCU combined platform

Use MQTT to control selections, puts all "intelligence" up into MQTT server, removing it from Arduino  
Adding WiFi requires appropriate interface. Wemos has single board with NodeMCU on it already (ish)  
Replace "music" machine with cheap Android Car Head Unit. Does it all, including BT, Spotify, YouTube, etc, etc

Start MQTT  
Start Adding WiFi  
Start Head Replacement

Purchased \$50 head unit from WISH/Banggood/ wherever  
Purchased cheap car speakers from WISH/Banggood/ wherever  
Low expectations surprised as was pretty good.  
Low expectations realised as they were pretty crap!

Replaced the Car Centric dashboard (Speed, SatNav, Reversing Camera, etc), with more standard Android (mobile type) desktop  
Decided whilst whilst audio quality was not a requirement, the constant disappointment from the speakers would annoy me too much

Bought 2" hand pair of Picoeters from eBay  
Quality much better!

Wired it all up and tested! All good.

Install all electronics and test

We're DONE!!!  
For now.....

Build the Enclosure  
Construct the framework  
Edge the frame with rubber strip to provide airtight speaker enclosure  
Order and cut the Polycarbonate (4mm sheets)  
Mirror (badly the first sheet)  
Decided it was so bad that created second sheet (better but not perfect!)  
Installed both to create magic mirror effect on matrix (nice)  
Wasp entire frame in 6mm flexo ply (lovely stuff!)  
Lots of sanding and vanishing  
Finalised and filled front panel

Use Pi as MQTT/NodeRed server  
Install Mosquitto  
Learn NodeRed!  
Build Flows and Dashboard in NodeRed to enable lights to be controlled by sending out MQTT commands  
Learn JSON!  
Learn Javascript

Dashboard Created and MQTT messages being sent out

Will smarten this up to enable the NodeMCU module to be used generically (i.e. configurable MQTT server/port/etc and WiFi config details). Currently hardcoded into the config.

Port the Arduino Mega code into the Wemos Mega (no changes)  
Rip out all the menu system  
Rip out the RTC code (will use MQTT to provide time if required)

Add code to read Serial3 (from NodeMCU) and parse the message  
Tested MQTT messages generated from NodeRed dashboard received by Mega (via NodeMCU) and correct light patterns/colours changed

Try and work out why its not working!!! I can see NodeMCU sending JSON, but appears in Mega as rubbish  
Whilst the obvious Baud rates were ok, 9600, still got corruption. Reduced the BAUD down to 1200 which made things better, however still having problems with some of the longer "topic" strings. Seems I was on the cusp of the serial port timeout, so amply where ok, but a couple of messages pushed it over. Updated the port timeout and reliability ensued. Still cannot understand why the Baud rate needs to be so low, not a problem for this app, but not very good!!! Especially as on the Wemos, the Serial line is physically internally routed from the NodeMCU over short PCB copper line (not even jumper cables). Hohum....

Upset the timeout and lowered the Baud and seems to be working now

Wired it all up and tested! All good.

Install all electronics and test

We're DONE!!!  
For now.....

0:22