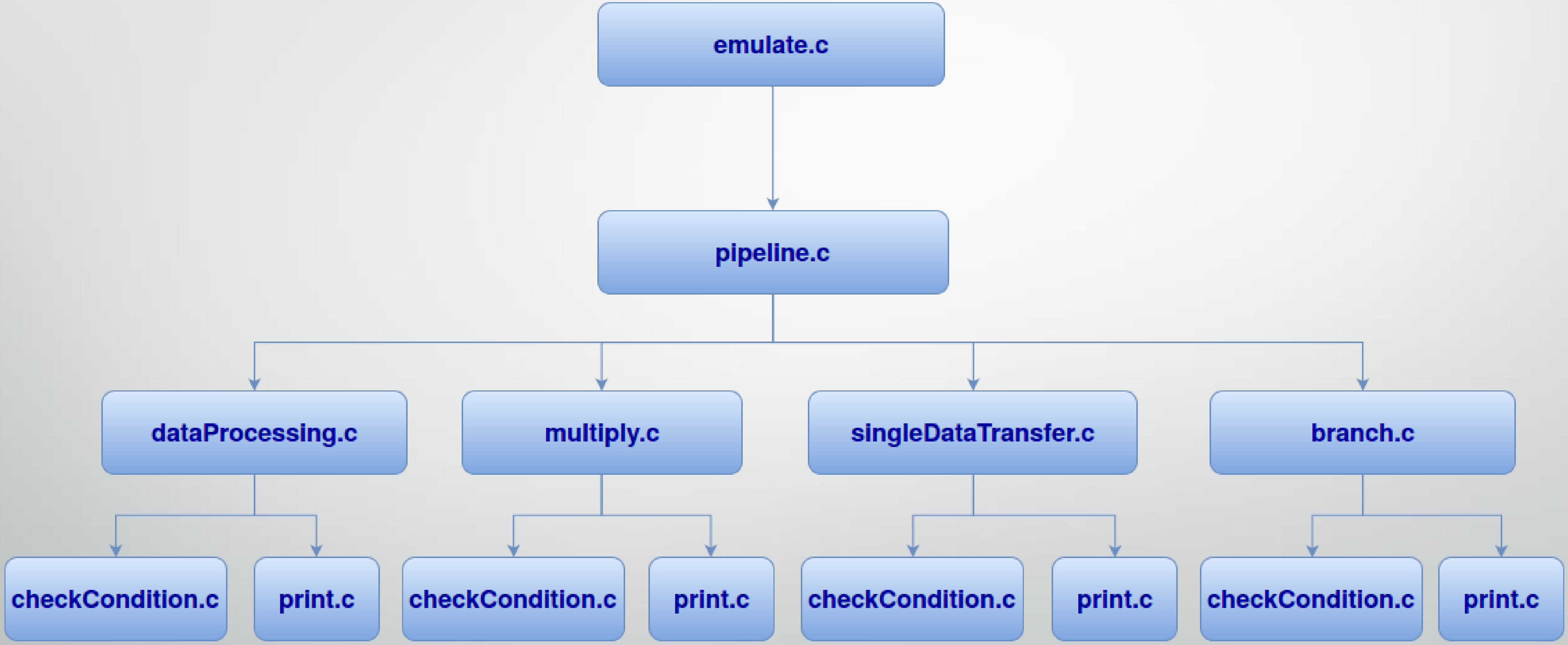
# Arm11-Group3

Andrew Pearcy, Marta Ungureanu, Oana Ciocioman, Maurizio Zen

# Emulator

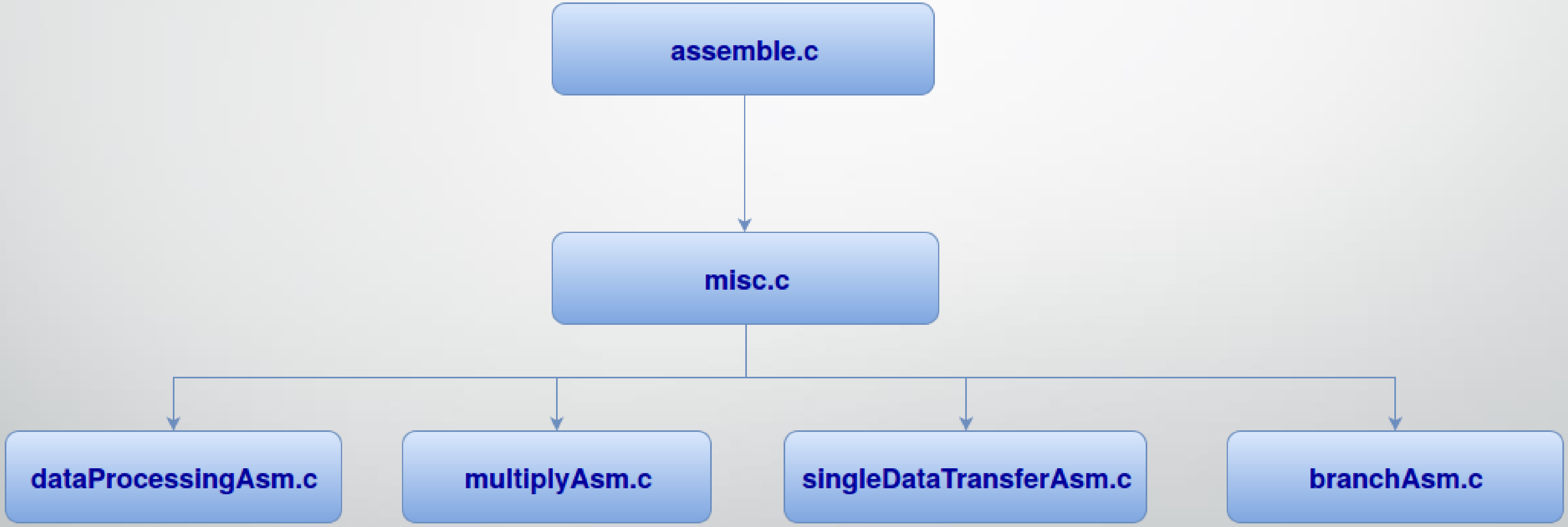

emulate.c
pipeline.c
dataProcessing.c
multiply.c
singleDataTransfer.c
branch.c
checkCondition.c
print.c
checkCondition.c
print.c
checkCondition.c
print.c
checkCondition.c
print.c

# Emulator - Challenges

- Synchronizing the pipeline
- Setting flags in Data Processing
- Optional instructions – special values
- Blank lines issue

http://www.preact.co.uk/media/images/media/challenges-sales.jpg

# Assembler



assemble.c

misc.c

dataProcessingAsm.c

multiplyAsm.c

singleDataTransferAsm.c

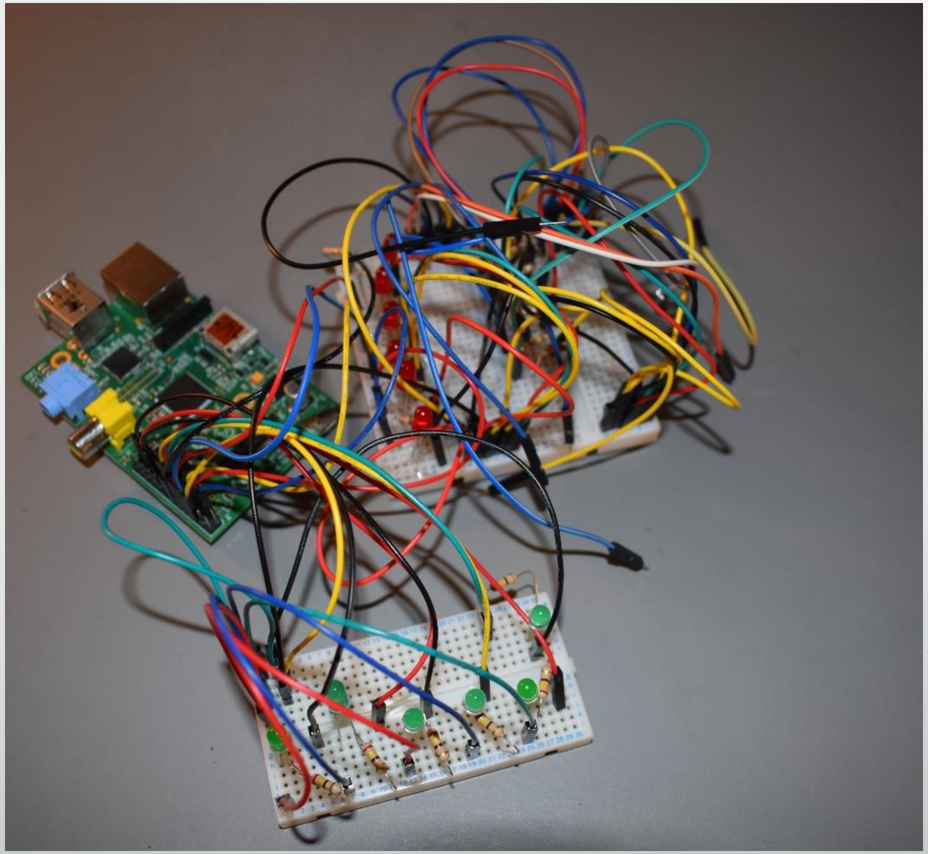
branchAsm.c

# Assembler - Challenges

- Differentiating between labels and instructions
- Data Processing:
  - 8-bit instruction
  - Shifted register
- Single Data Transfer: immediate value for load
  - 8-bit value treated as move
  - Encode the value separately
- Optional instructions

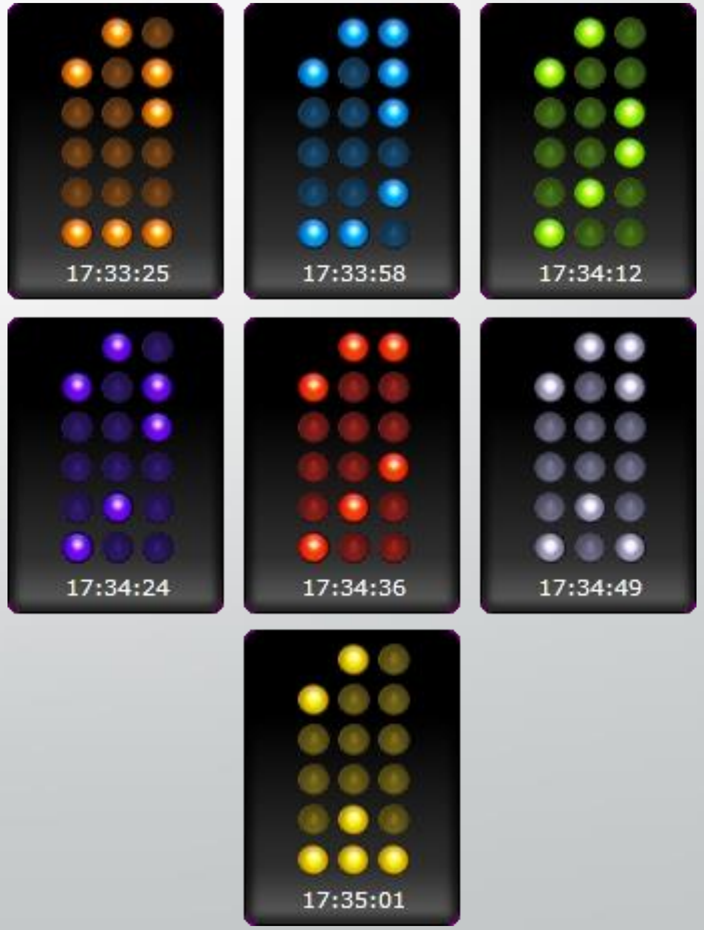
https://www.tnooz.com/wp-content/uploads/2011/11/angry-computer.jpg

# Our Extension



# What is a Binary Clock?

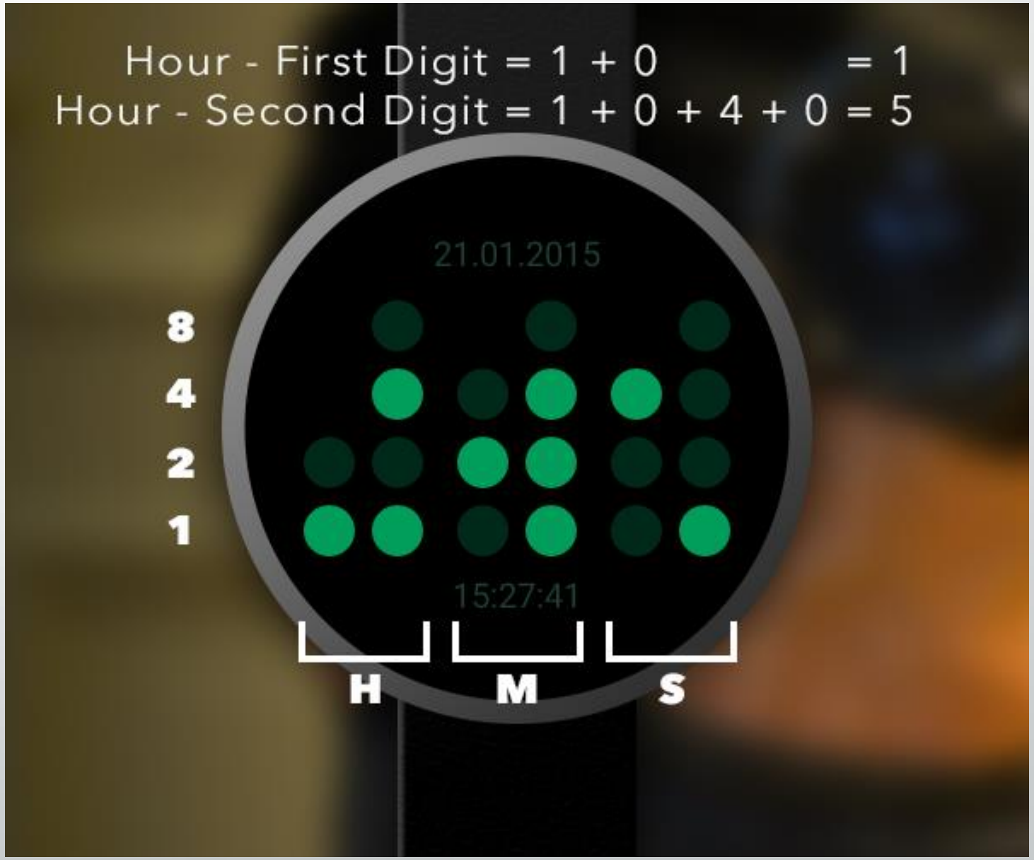
- A binary clock is a clock that displays the time of the day in a binary format.
- We represented the powers of two using LED's



https://nickstips.files.wordpress.com/2010/10/colors.png

# First Approach

- HH:MM:SS format
- Each column represents a digit

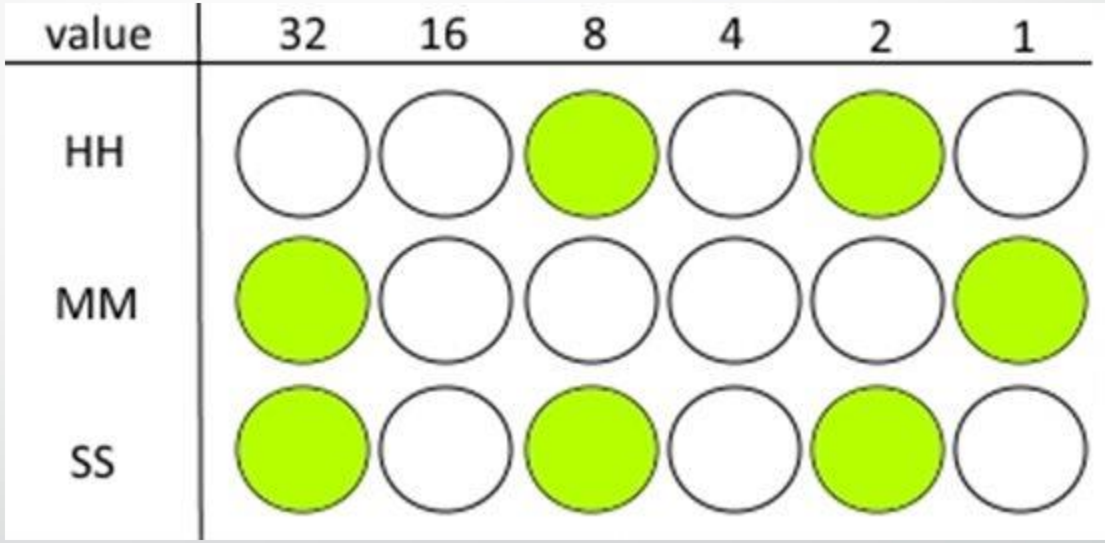


Hour - First Digit = 1 + 0 = 1
Hour - Second Digit = 1 + 0 + 4 + 0 = 5

https://lh3.ggpht.com/X9IaQzNsMwfCs7Uk9VBZWynlJzga6uBfFwybWdJ7vlVC_MUiOAPwVYm7xuoBs6CVZmM=h900

# Second Approach

- Three Rows:
  - Hours
  - Minutes
  - Seconds

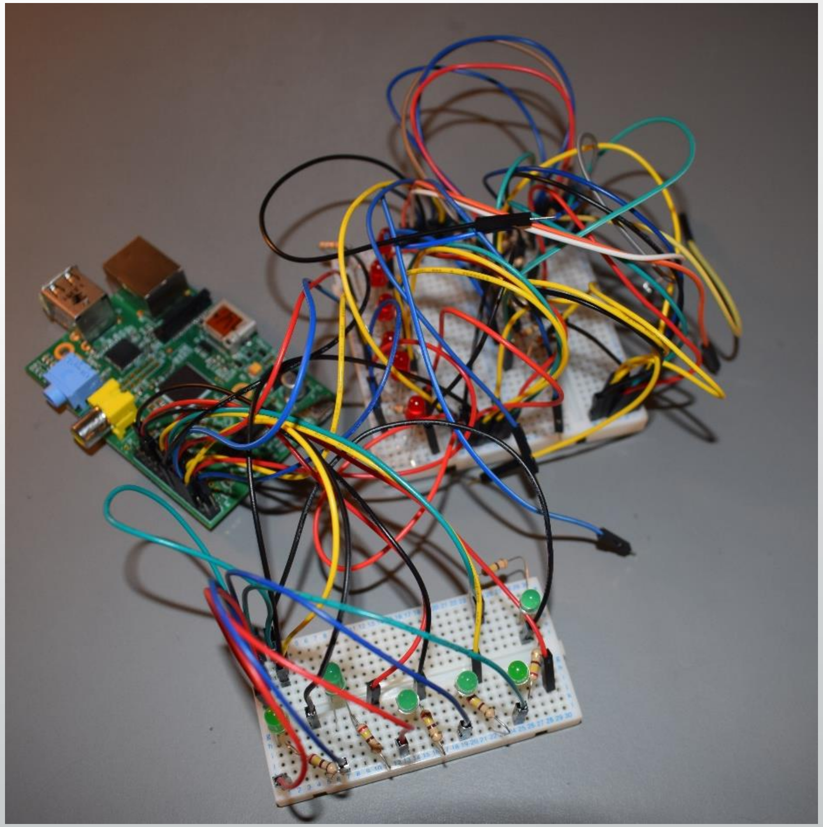


| value | 32 | 16 | 8 | 4 | 2 | 1 |
|---|---|---|---|---|---|---|
| HH | ○ | ○ | ● | ○ | ● | ○ |
| MM | ● | ○ | ○ | ○ | ○ | ● |
| SS | ● | ○ | ● | ○ | ● | ○ |

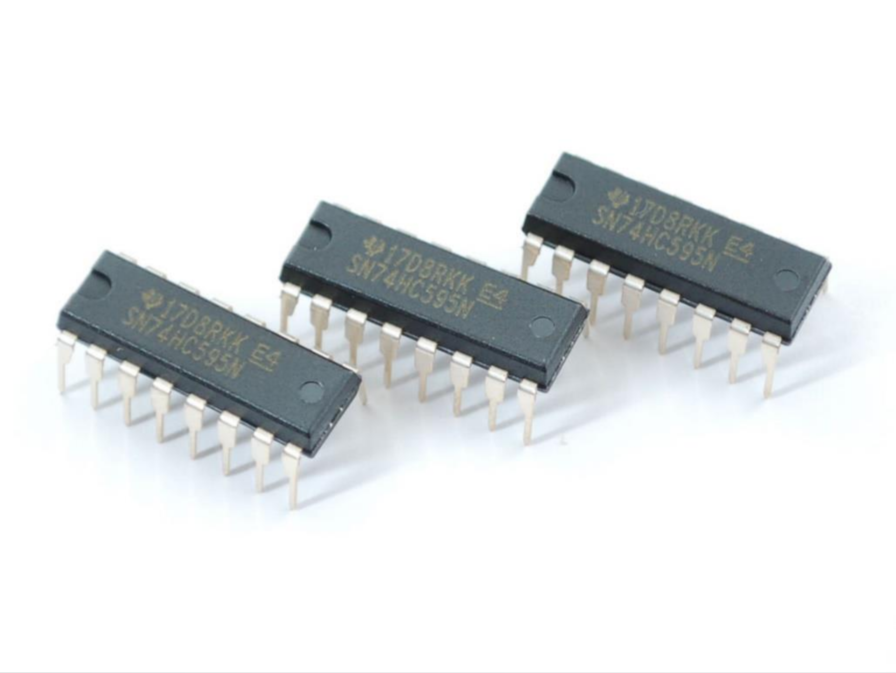
http://social.technet.microsoft.com/wiki/cfs-filesystemfile.ashx/__key/communityserver-components-imagefileviewer/communityserver-wikis-components-files-00-00-00-00-05/1488.hour.jpg_2D00_550x0.jpg

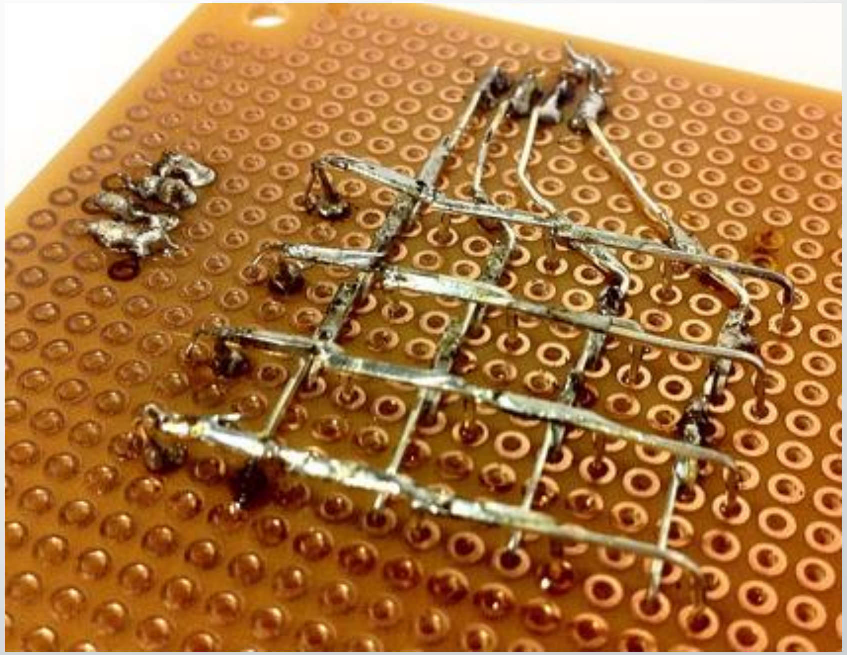
# Prototype 1

- 17 LED’s using all 17 GPIO pins



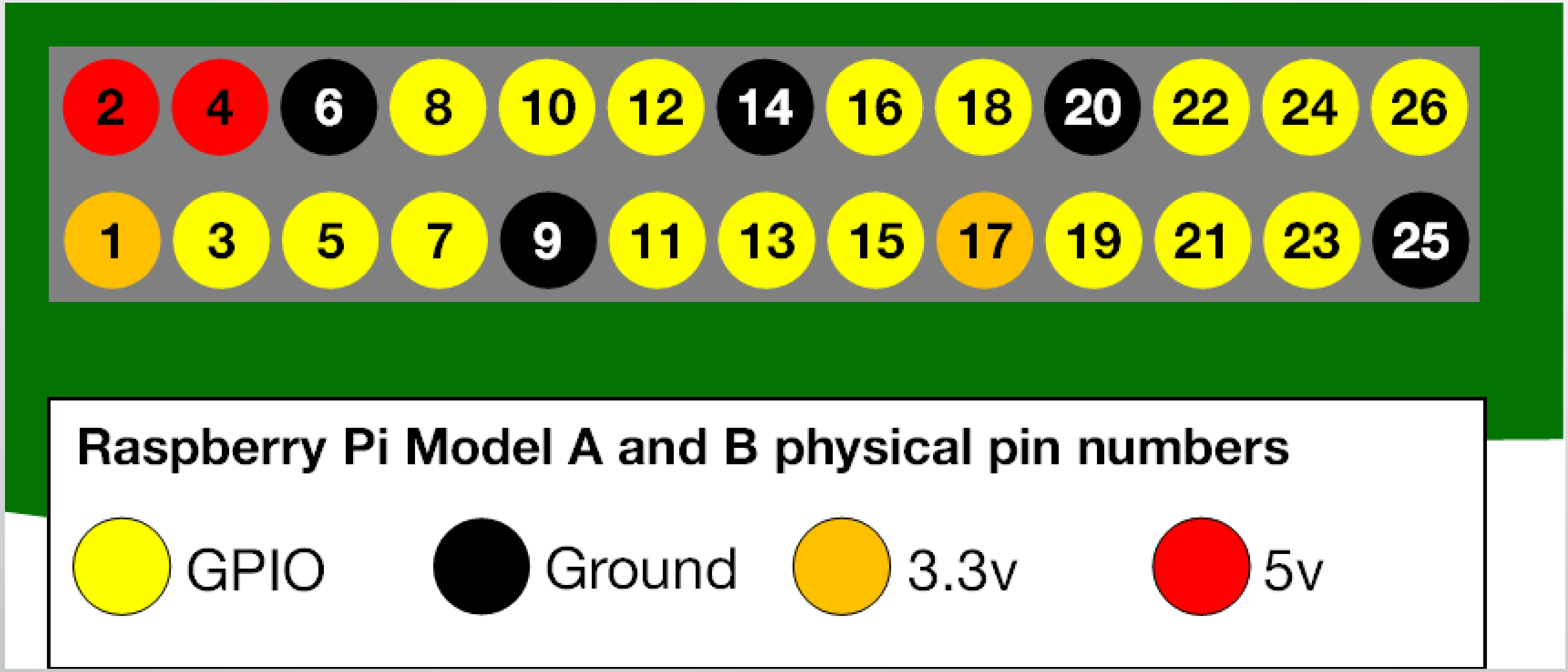
# Shift Registers VS LED matrix





http://staticjolt.com/wp-content/uploads/2014/12/450-00.jpg     http://www.appelsiini.net/attachments/4x4-bottom.jpg

# Problem – All 17 GPIO Pins Used



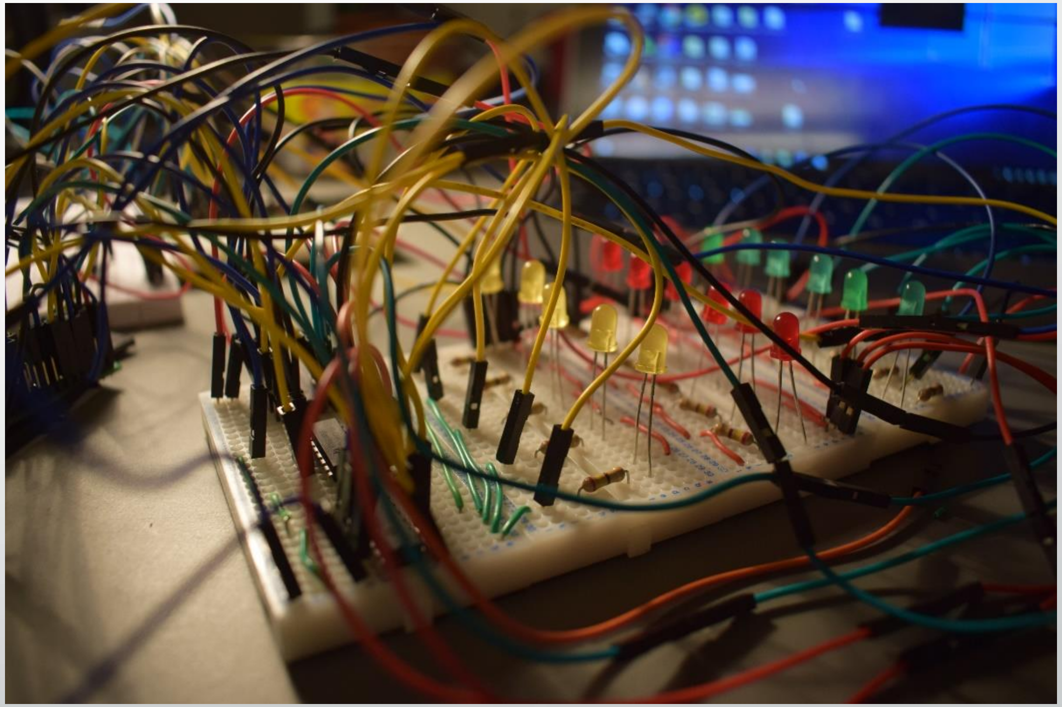
| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 | 18 | 20 | 22 | 24 | 26 |
| 1 | 3 | 5 | 7 | 9 | 11 | 13 | 15 | 17 | 19 | 21 | 23 | 25 |

Raspberry Pi Model A and B physical pin numbers

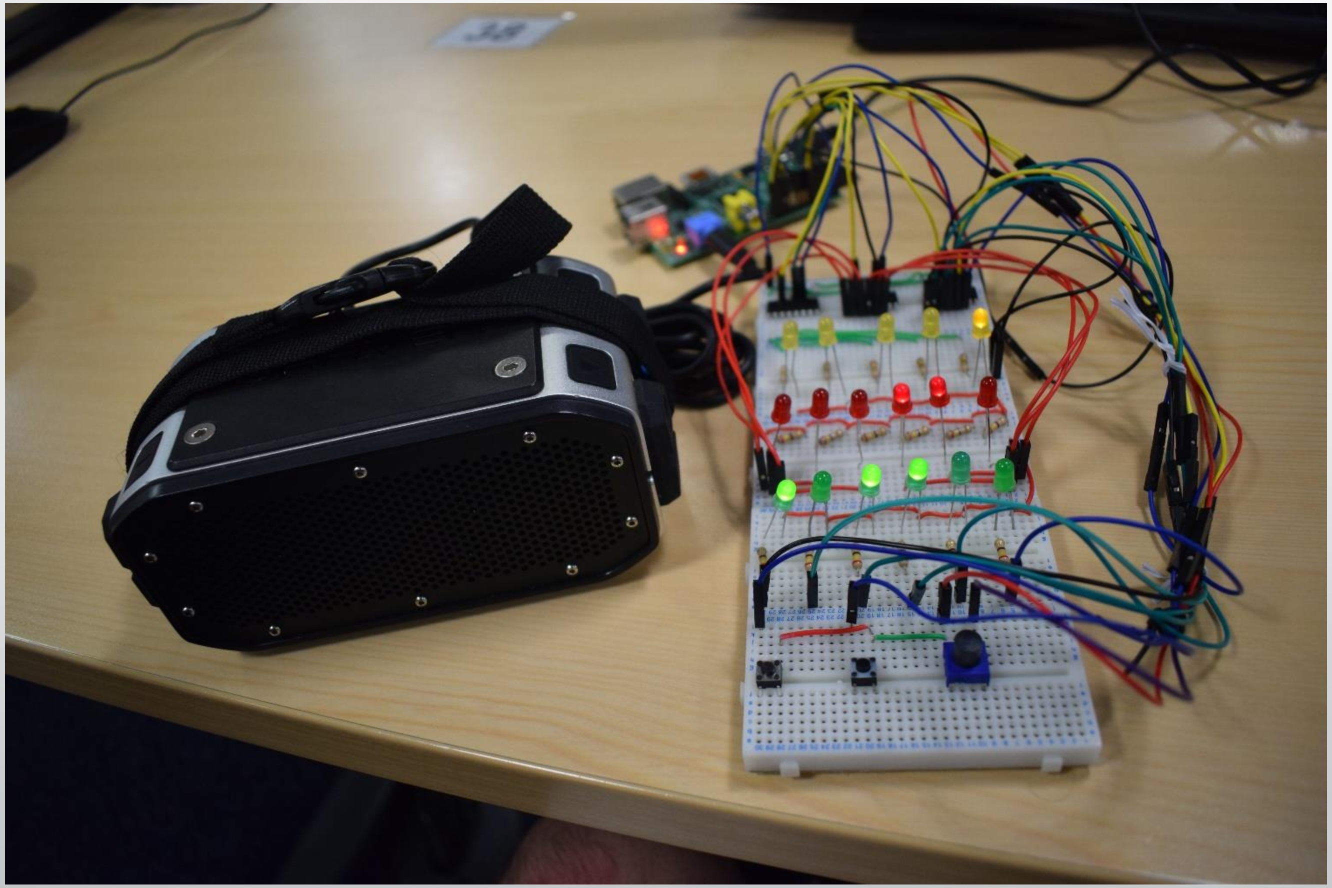
https://www.raspberrypi.org/documentation/usage/gpio/

# PROTOTYPE 2

- Utilisation of shift registers
- allowed additional inputs.
- Freed GPIO pins to be used for
- button inputs



# Finished Prototype



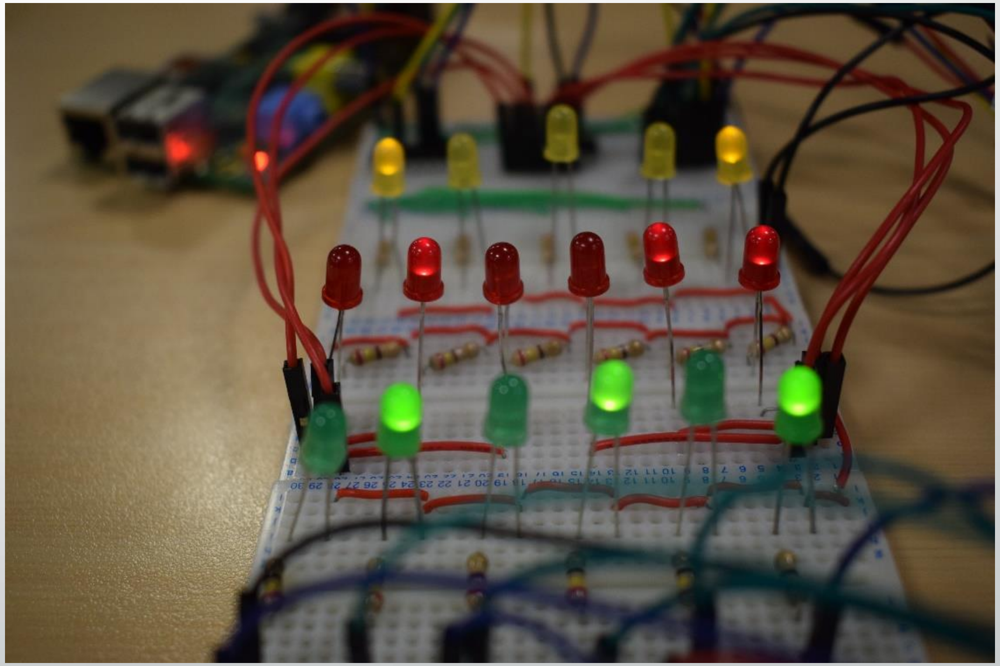
# Implementation

- WiringPi
- First attempt: 17 LEDs, one LED connected to each GPIO pin
  - Read system time and represent by writing HIGH/LOW to each pin
- Prototype improved with shift registers
  - Sending data serially, which is then executed in parallel by the registers
  - The only modification was how data was transmitted to the LEDs

http://www.theflippedclassroom.es/wp-content/uploads/2014/07/Collaboration-team-work.jpg.pagespeed.ce_.Jf_WSBWV8u-300x203.jpg

# Implementation Part 2 – Final Product

- System time problematic
  - Added buttons to set the time and change the mode from CLOCK_MODE to SET_TIME_MODE
- Alarm
  - Used the same buttons, added a new mode, SET_ALARM_MODE
- Issues
  - How often we read the buttons
  - Initially, the clock started only after setting the alarm resulting in lost seconds/minutes



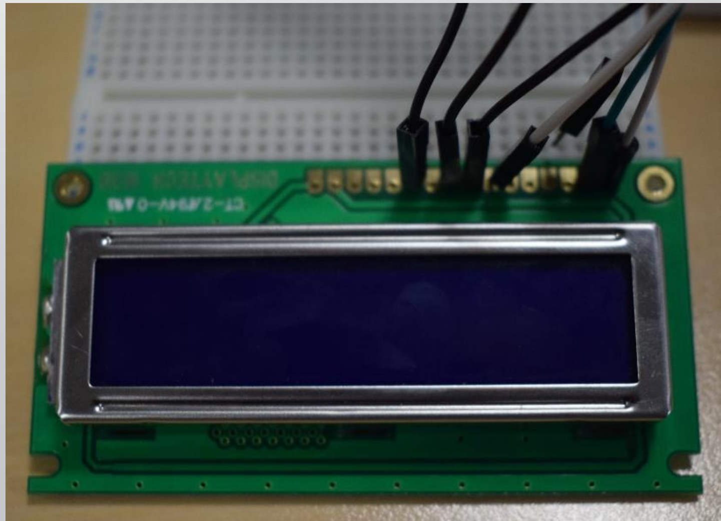
# The Alarm

- Set and stopped using the mode button
- A system command plays the audio file in the background, in a child process
- Child process killed when mode button is held down
- The clock continues to display the time as the alarm is on

http://images.mydoorsign.com/img/lg/K/Alarm-Sound-Security-Sign-K-4713.gif

# Further possible extensions

- Allowing the user to set multiple alarms, by storing the alarms in a dynamically allocated array
- Adding a snooze button (pretty straightforward)
- Using a display to output the mode the clock is in/the time in decimal
- Remembering/displaying the date (using the display)
- Auto changing from summer time to winter time



# Testing our Alarm Clock

- Primarily physical testing
- Alarm tests hard coded into the implementation

http://techno-bliss.com/wp-content/uploads/2015/04/software-Testing.png

# Reflection

- What we did well:
  - Writing common functionality before splitting the work
  - Constant communication and almost daily meetings
  - Time management
  - Collaboration when debugging
- What we need to improve on:
  - Utilization of Git

http://www.specialeducationadvisor.com/wp-content/uploads/2012/10/looking-back.jpg

Questions?