Implementing a top-interface for the Pi 3B

Group 24

Introduction to Embedded Systems

1 November 2019

Overview

Project Introduction

Overview
    data extraction from the kernel interface
    top-interface for the LED

programs for load-simulation

future work

Introduction

- ▶ data extraction from the kernel interface
- ▶ 'top'-interface for the LED
- ▶ programs for load-simulation

How can we acquire stats from the cpu?

▶ pseudo filesystem proc as the kernel interface

▶ parsing files

```
fopen("/proc/stat");
fgets(buffer);
strtok(buffer);
```

How can we display the information on the LED?

There are multiple ways of displaying this information on the 8x8 grid. We picked a simple approach.

- ▶ calculate the load (in %) for every cpu
- ▶ set up threshold percentages for each of the 8 lights per row
- ▶ iterate through the cpu data and light the corresponding row accordingly

How can we display the information on the LED?

```
for ( size_t i = 0; i <= MAX_CPU; i++) {
    perc = cpulist [ i]−>perc_use ;

    if ( perc > 0)
      set_led ( i , 0, RGB565_GREEN ) ;
    if ( perc > 12.5)
      set_led ( i , 1 , RGB565_GREEN ) ;
    [ . . . ]
}
```

How can we check if our program reacts to changing workload?

We have written a few extra programs that utilize forks and some random calculations to change the workload.
This was intended to be controlled by the joystick, but discarded as the change in the workload is no goal of the program itself.

```c
int main(void) {
  fork();
  size_t i = 0;
  while (1)
    i++;
}
```

There are still many improvements possible, including:

- ▶ display of cpu-data on a webserver
- ▶ additional problem: proper save of acquired data
- ▶ more datasets (e.g. memory load, swap)

Interested? Check out the codebase

https://github.com/Lordakius/pi-senseHAT-top