

F28HS Hardware-Software Interface

Lecture 0: Overview

Lecturers

- Hans-Wolfgang Loidl
 - EM G51
 - H.W.Loidl@hw.ac.uk
 - X 3421
- Alistair C. McConnell
 - EM G49
 - Alistair.McConnell@hw.ac.uk
 - x 3430

Aims

- To gain an understanding of low-level, **hardware-oriented and systems programming**.
- To develop skills in **resource-conscious programming**.
- To develop programming skills in such languages (**C and ARM Assembler**).

Syllabus

- Low-level, assembler programming
- Low-level, C programming
- Advanced computer architecture issues impacting software performance (caches, multi-cores, etc)
- Operating system interfaces for low-level software

Syllabus

- Operating system concepts such as device handling, interrupts, BIOS etc
- Embedded systems programming
- **Resource-conscious programming** techniques (memory, performance; programming techniques, tools, monitoring)

Subject mastery

- Critical understanding of computer architecture concepts and their performance implication for low-level software.
- Detailed theoretical and practical understanding of hardware and operating system concepts, interfacing to low-level software.
- Ability to develop efficient, resource-conscious code, interfacing to hardware components.
- Practical skills in low-level, systems programming, with effective resource management.

Personal abilities

- Ability to articulate system-level operations and to identify performance implications of given systems

Organisation

- Systems Programming Track
 - In Odd weeks (1,3,5,7,9,11)
 - By Hans-Wolfgang Loidl
 - On embedded systems programming, using the Raspberry Pi
- Programming Languages Track:
 - In even weeks (2,4,8,10,12)
 - By Alistair C. McConnell
 - On C & assembly language programming

Timetable

- Monday 9:00 in **LT3** (f2f)
 - Tutorial
- Monday 10:00-11:00 EM 2.50 (f2f)
 - Lab (for **surnames A-J**)
- Monday 11:00-12:00 EM 2.50 (f2f)
 - Lab (for **surnames K-Z**)
- Tuesday 17:00-18:00 JW2 (f2f)
- Friday 11:00-12:00 PG G.01(f2f)

Effort Hours

Activity	Hours
Tutorial	1 hour (scheduled slot Mon 9:00am; sync, face-to-face)
Lab	2 hours (scheduled slot Mon 10:00am; sync, face-to-face)
Lectures	2 hours (scheduled slots on Tue and Fri; sync, face-to-face)
Coursework	3 hours (in your own time, ask for help on GitLab, asynchronous)
Additional Reading	1 hours (in your own time)
Watching additional supportive HOWTOs	1 hour (in your own time, asynchronously, online)

Assessment

- No exam
- **Coursework: 100%**
 - **CW1**: C programming exercise
 - **CW2**: Systems programming exercise on the Raspberry Pi, using C and ARM Assembler

Assessed coursework

- Alistair McConnell
 - CW1: Programming in C exercise
 - Weighting: 50%
 - Individual
 - distributed: week 3 (Feb 2nd)
 - **submission: week 7 (Feb 29th)**
 - feedback: week 10 (Mar 21st)

Assessed coursework

- Hans-Wolfgang Loidl
 - CW2: Systems programming on the Raspberry Pi
 - Weighting: 50%
 - In pairs
 - distributed in Week 8 (Mar 5th)
 - **submission in Week 12 (Apr 4th)**
 - feedback in Week 15 (Apr 25th)
 - plus an **Assessed Lab in Week 5**
 - Bringing together C & assembler programming, applied to systems programming

Raspberry Pi

- course based around Raspberry Pi 2/3 computer; (RPi 4 possible)
- single board system
- 900 MHz quad-core ARM Cortex-A7
- 1 GB RAM
- runs Raspberry Pi OS (previously called Raspbian) variant of Linux
- BCM 2835 General Purpose I/O (GPIO) chip for hardware/software experiments

Raspberry Pi

- Raspberry Pi 2/3 + hardware kit
- available on loan from Computer Technician
- **Arrange a pick-up slot for your kit by email to: a.c.hurt@hw.ac.uk**
- plug in to monitor/mouse/keyboard for Linux desktops in EM 2.50
 - KVM (keyboard-video-mouse) switch
- **Kits must be returned by end of Week 13 (April 12th) or course marks will be withheld!**

Raspberry Pi

GPIO

red light

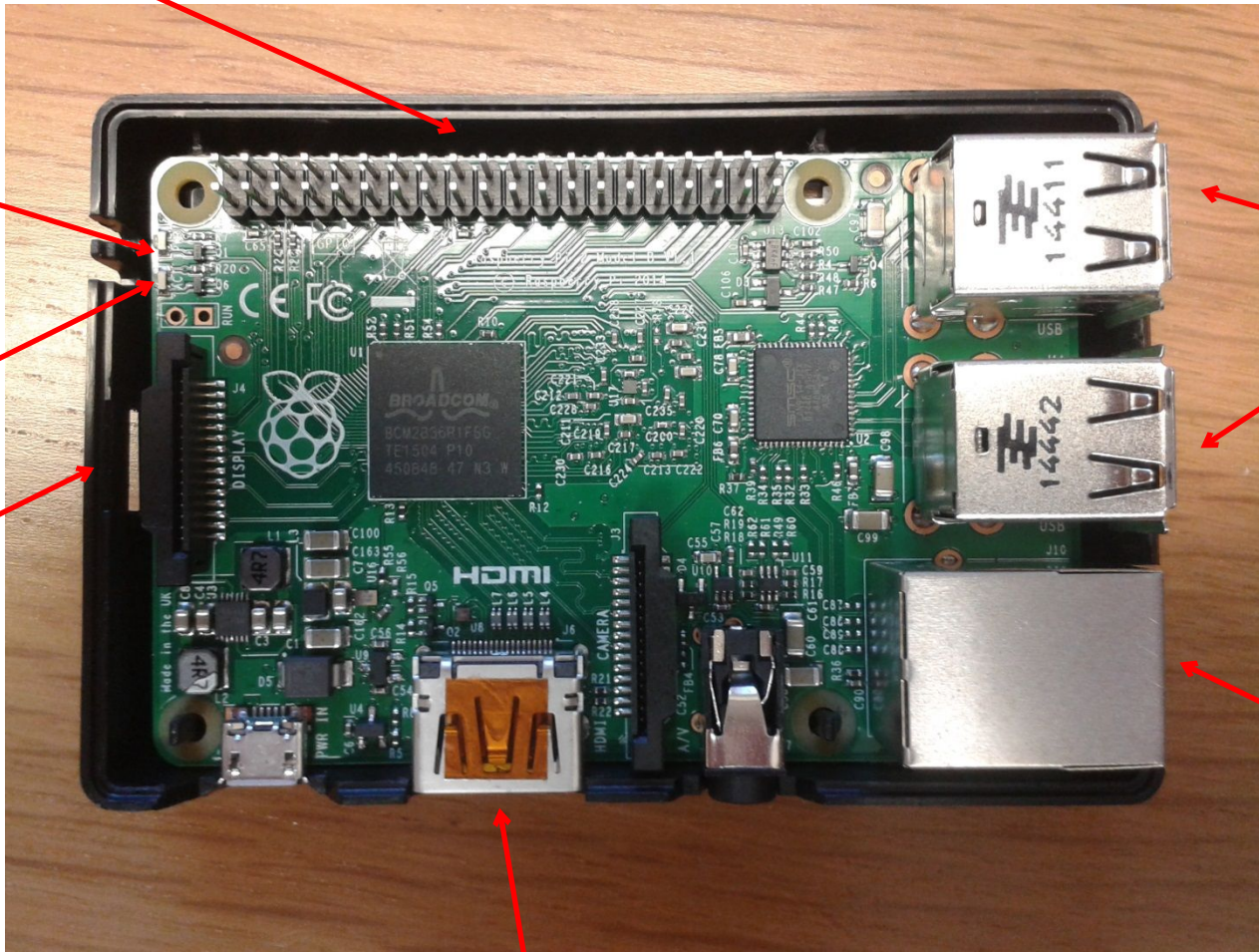
green light

micro SD with Raspbian

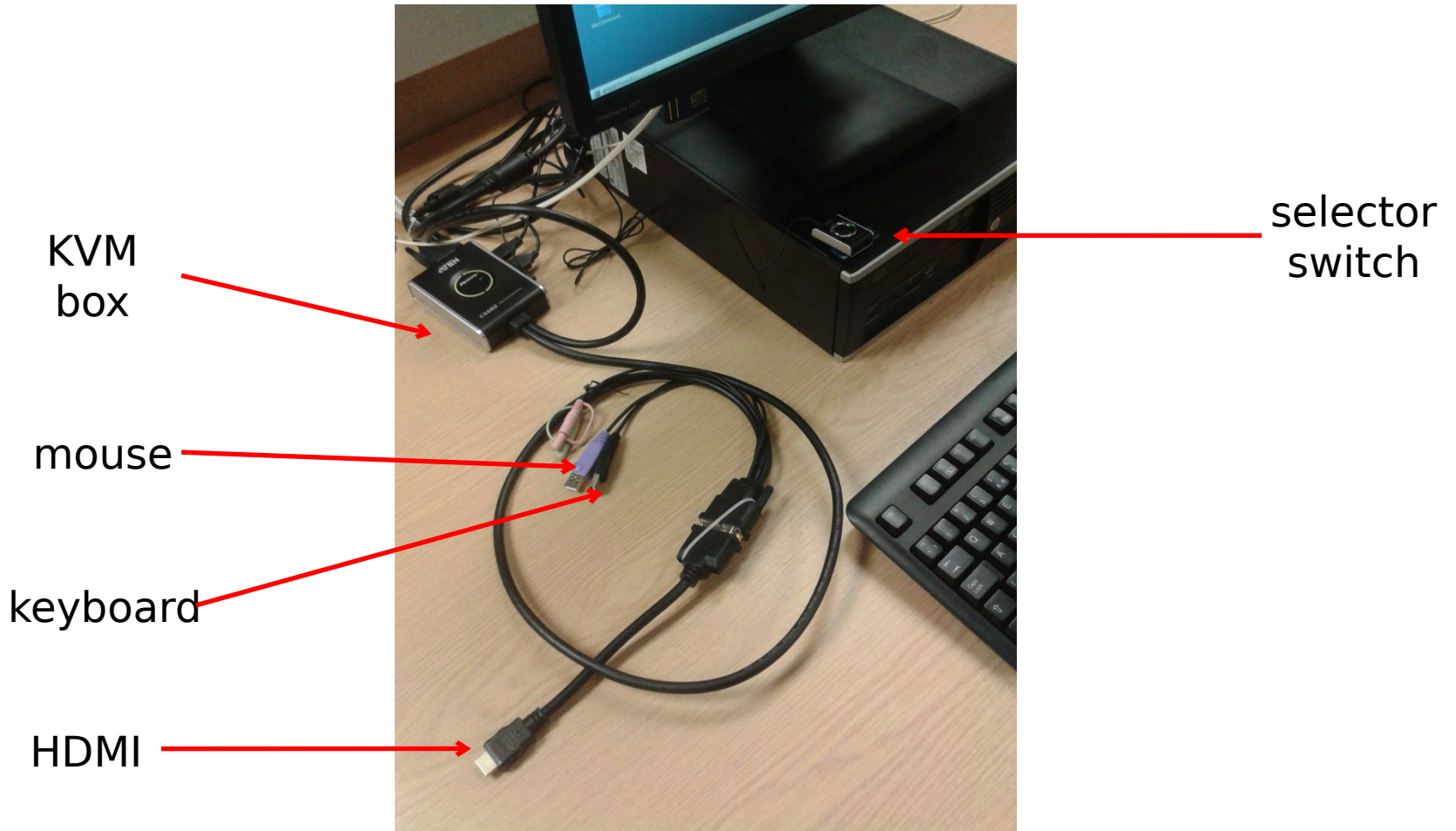
USB

ethernet

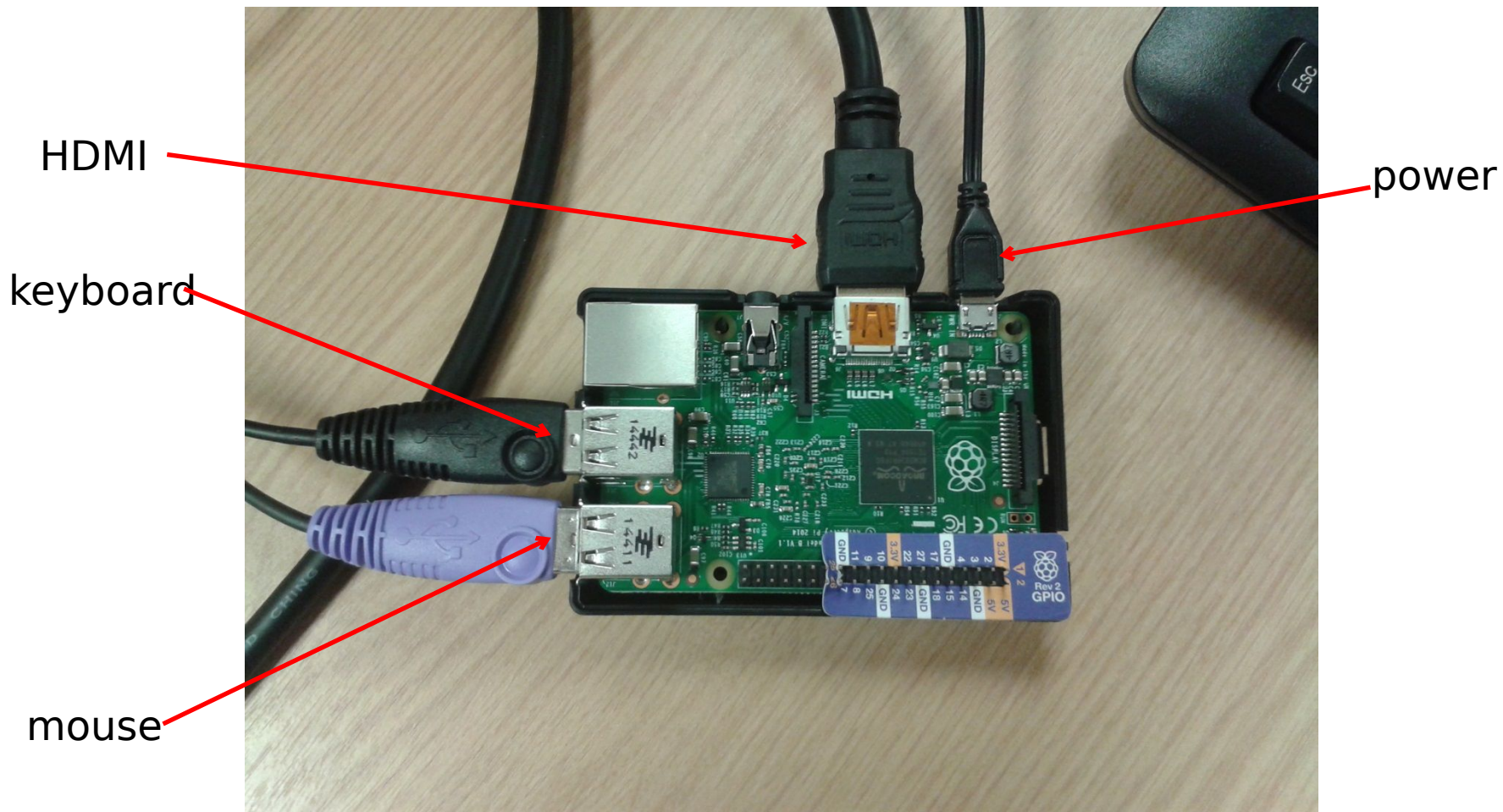
HDMI



Raspberry Pi



Raspberry Pi



Raspberry Pi

- plug in mouse, keyboard & HDMI
- push KVM button
- login: pi
- password: raspberry
- to run GUI: startx

