

# F28HS Hardware-Software Interface

Lecture 0: Overview

# Lecturers

- Hans-Wolfgang Loidl
  - EM G51
  - [H.W.Loidl@hw.ac.uk](mailto:H.W.Loidl@hw.ac.uk)
- Tess Watt
  - EM 1.55
  - [T.Watt@hw.ac.uk](mailto:T.Watt@hw.ac.uk)



# Key Resources

- Canvas page for F28HS
  - All core resources for this course
- Gitlab-student repo
  - For sample sources and CW templates
- HaWo's F28HS course info page
  - For quick summary

# 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, **ARM 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)

# Learning Outcomes

- Demonstrate an understanding of modern computer architecture concepts and their interactions
- Identify performance implications of computer architecture concepts for low-level software.
- Develop efficient, resource-conscious software, interfacing to hardware components
- Develop low-level systems programming software with efficient resource management
- Demonstrate a theoretical and practical understanding of hardware interfaces in low-level software

# Organisation

- We deliver the course in **two tracks**.
- **Systems Programming Track:**
  - Tutorials in odd weeks (1,3,5,7,9,11)
  - By Hans-Wolfgang Loidl
  - On embedded systems programming, using the Raspberry Pi
- **Programming Languages Track:**
  - Tutorials in even weeks (2,4,8,10,12)
  - By Tess Watt
  - On C & ARM assembler language programming

# Timetable

- Monday 10:00 in LT1
  - Tutorial
- Monday 11:00-12:00 EM 2.50
  - Lab (for **surnames A-J**)
- Monday 12:00-13:00 EM 2.50
  - Lab (for **surnames K-Z**)
- Tuesday 13:00-14:00 LT1
- Tuesday 14:00-15:00 LT1

# 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; 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 (40%)**: C programming exercise
  - **CW2 (60%)**: Systems programming exercise on the Raspberry Pi, using C and ARM Assembler

# Assessed coursework

- Tess Watt
  - CW1: Programming in C exercise
    - Weighting: 40%
    - Individual
    - distributed: week 3 (Jan 26<sup>th</sup>)
    - **submission: week 7 (Feb 24<sup>th</sup>)**
    - feedback: week 10 (Mar 17<sup>th</sup>)

# Assessed coursework

- Hans-Wolfgang Loidl
  - CW2: Systems programming on the Raspberry Pi
    - Weighting: 60%
    - Individual
    - distributed in Week 7 (Feb 23<sup>th</sup>)
    - **submission in Week 11 (Mar 24<sup>th</sup>)**
    - feedback in Week 14 (Apr 17<sup>th</sup>)
  - plus an **Assessed Lab in Week 5**
  - Bringing together C & assembler programming, applied to systems programming

# AI policy

- **Level 1 (“GenAI language support”) of the HWU genAI policy applies to this course:**
  - Where English is not your first language you may use GenAI to translate the assessment instructions or other relevant content.
  - You may not use GenAI to translate your assessment answers, you must write them in English as HW requires all work to be completed in English and to be your own work.
  - No GenAI generated content is permitted in your assessment submission.
- **The code you submit must be your own work.** If some text or code in the coursework has been taken from other sources you must give a clear reference.

# Raspberry Pi

- course based around Raspberry Pi 2 or 3 computer; (RPi 4 possible, with code change)
- single board system, minimum:
  - 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

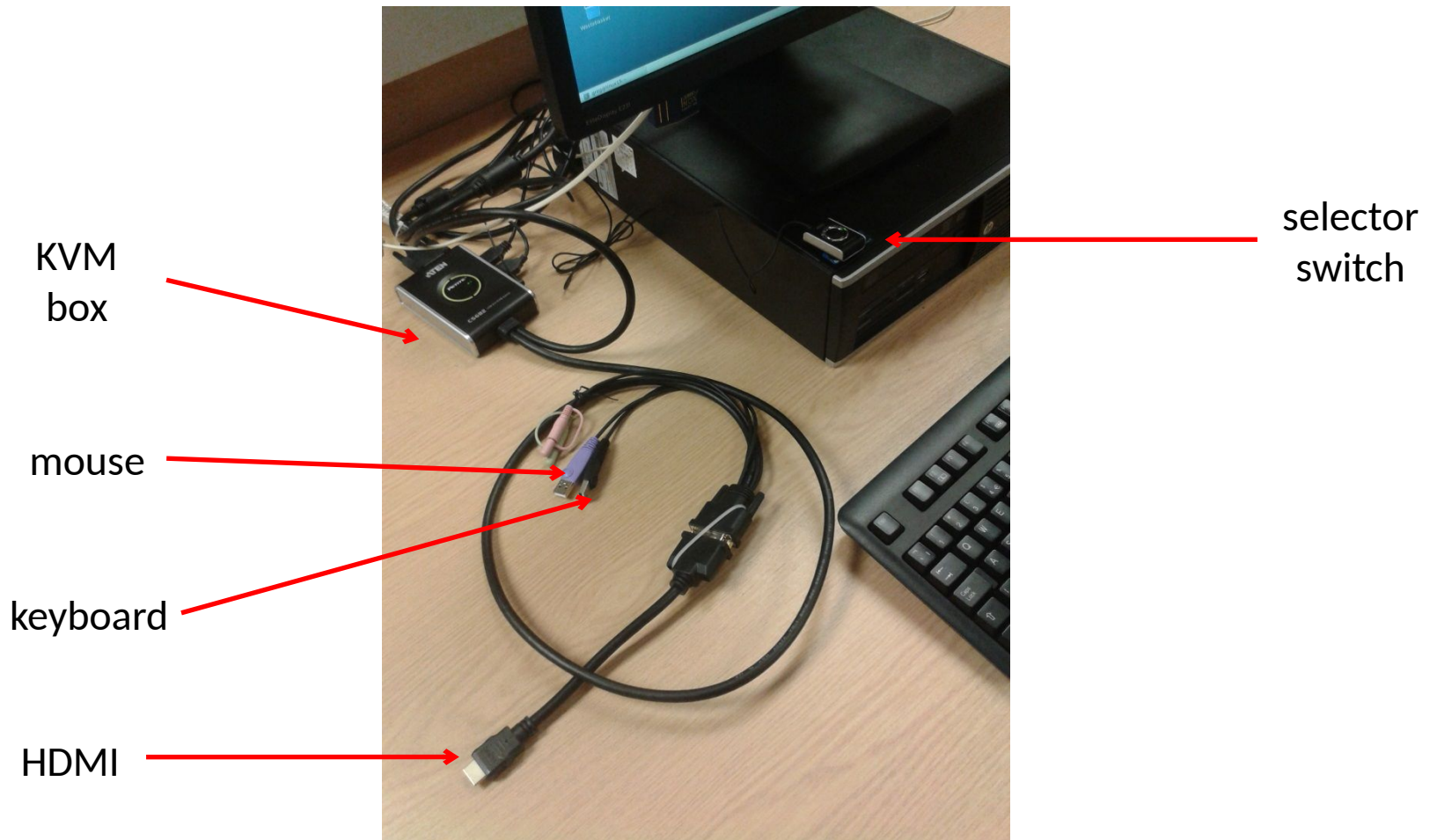
# Raspberry Pi

- Raspberry Pi 2 or 3 + hardware kit
- available on loan from IS Services
- **Pick-up your RPi kit on a weekday btw 9:00am-4:30pm from AM01 (Alan McTernan Building)**
- 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 10<sup>th</sup>) or course marks will be withheld!**

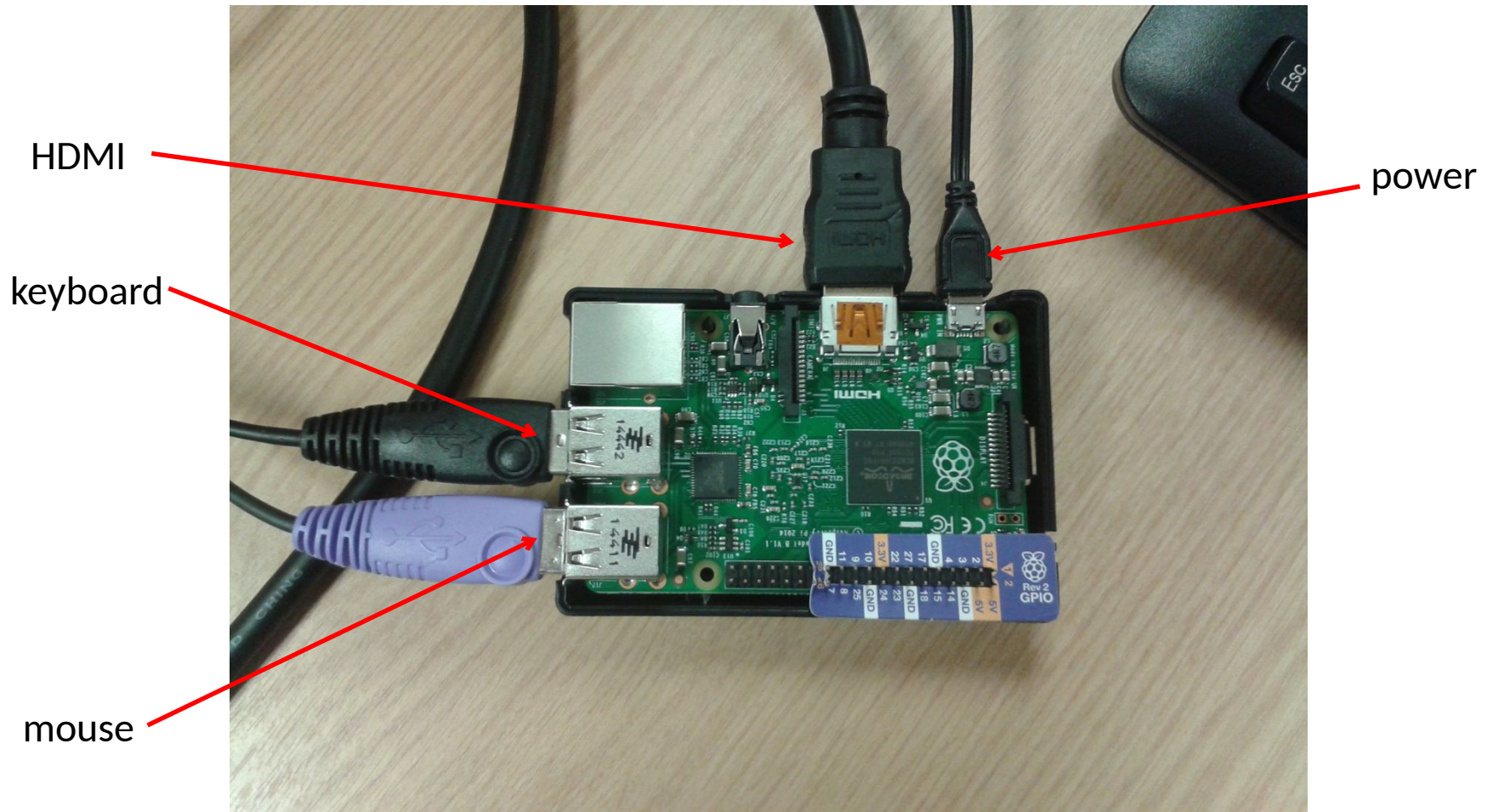
# Raspberry Pi



# Raspberry Pi



# Raspberry Pi



# Raspberry Pi

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

