EMLi portfolio assessment guide

Version 2024-04-16v3, Kjeld Jensen kjen@sdu.dk

Introduction

This document constitutes the correction guide for the Embedded Linux (EMLI) 5 ECTS course, MSc. level “individual presentation portfolio" which must be graded according to the 7-point grading scale.

Learning objectives

Knowledge

- Linux architecure
- Embedded Linux
- Distributed embedded linux computers wired and wireless networks
- I/O interfacing to sensors and actuators

Skills

- Integrate embedded Linux computers into relevant applications
- Apply appropriate communication and message passing architectures for data exchange and remote management, use version control systems

Competences

- Leverage and adapt existing embedded Linux design patterns to new applications.
- Create generic embedded Linux computers that can be combined to solve specific cyber-physical problems

Karakterskala/Point grading scale

12 Den fremragende præstation (ECTS A)
Gives for den fremragende præstation, der demonstrerer udtømmende opfyldelse af fagets, fag- eller uddannelseselementets mål, med ingen eller få uvæsentlige mangler.

Excellent
For an excellent performance displaying a high level of command of all aspects of the relevant material, with no or only a few minor weaknesses.

10 Den fortrinlige præstation (ECTS B)
Gives for den fortrinlige præstation, der demonstrerer omfattende opfyldelse af fagets, fag- eller uddannelseselementets mål, med nogle mindre væsentlige mangler.

Very good
For a very good performance displaying a high level of command of most aspects of the relevant material, with only minor weaknesses.

7 Den gode præstation (ECTS C)
Gives for den gode præstation, der demonstrerer opfyldelse af fagets, fag- eller uddannelseselementets mål, med en del mangler.

Good
For a good performance displaying good command of the relevant material, but also some weaknesses.

4 Den jævne præstation (ECTS D)
Gives for den jævne præstation, der demonstrerer en mindre grad af opfyldelse af fagets, fag- eller uddannelseselementets mål, med adskillige væsentlige mangler.

Fair
For a fair performance displaying some command of the relevant material, but also some major weaknesses.

02 Den tilstrækkelige præstation (ECTS E)
Gives for den tilstrækkelige præstation, der demonstrerer den minimalt acceptable grad af opfyldelse af fagets, fag- eller uddannelseselementets mål.

Adequate
For a performance meeting only the minimum requirements for acceptance

00 Den utilstrækkelige præstation (ECTS Fx)
Gives for den utilstrækkelige præstation, der ikke demonstrerer en acceptabel grad af opfyldelse af fagets, fag- eller uddannelseselementets mål.

Inadequate
For a performance which does not meet the minimum requirements for acceptance.

-3 Den ringe præstation (ECTS F)
Gives for den helt uacceptable præstation.

Poor
For a performance which is unacceptable in all respects.

Reference: https://www.retsinformation.dk/eli/lta/2022/1125

Individual presentation portfolio

The individual presentation portfolio contains:

1) **A report describing the project.** Report pages must be A4 format, single column and have 2 cm margins. Any pages beyond the stated maximum number of pages will not be read. Add images, sketches etc. as applicable. Page numbering must be present but no other headers or footers on the pages. Font must be Times New Roman, Liberation Serif or similar at 12pt. The report consists of two individual pdf documents described in the following:
   a) **a team document (11 pdf pages)** following the structure:
      - Front page: clearly stating project group number, group member names and SDU email addresses, and link to the git repository. No other report content at this page. This front page is included in the page count i.e. there are 10 pages left for the actual document.
      - Solution approach: Describe the high level solution approach, the overall architecture and design, including advantages and drawbacks. Use sketches and diagrams.
      - Solution description: Describe solutions to key tasks and concepts in the project.
      - Tests and results: Describe how the embedded system was tested and present the results of those tests.
      - Conclusion: Short conclusion summarizing the achieved solution and results.
   b) **an individual document (4 pdf pages)** following this structure:
      - Front page: clearly stating student name, project group number and SDU email address. No other report content at this page. This front page is included in the page count i.e. there are 3 pages left for the actual document.
      - Discussion: Compare and discuss the achieved solution, results, demonstration to the functional requirements. Discuss to what extent the solutions achieve the nonfunctional requirements, and if not, why not.
      - Conclusion: Summarize and outline how the achieved solution and results can be improved in future work.
2) **A team video** presenting the outcome of the project:
   a) The video will be a combination of illustrations, recorded video of the physical system and any interactions, relevant screen recordings etc.
   b) The video length must be maximum 90 seconds.
   c) The video must be formatted as MP4 encoded using H.264/AAC with a maximum file size of 25 Mbytes. If not, the video may not be included in the assessment. Use of `ffmpeg` to convert to this format is recommended, the following parameters appears to work well for different sizes and formats:
      - `ffmpeg -i input.mov -c:v libx264 -vf scale=-1:1080 -crf 23 -maxrate 2M -bufsize 4M output.mp4`
3) **A link to a git repository** containing data according to the functional requirements as well as all relevant scripts, configuration files etc. created by the team for solving the project.
   a) The git repository must provide a quick and intuitive access to the relevant material.

Functional requirements

The functional requirements for the wildlife camera are:

a) The wildlife camera must run Raspberry Pi OS

b) The wildlife camera must have a shell script `take_photo.sh` that when executed, takes a photo using the Raspberry Pi camera and saves it to disk. The filename must be formatted as "`143557_045.jpg`" for local summer time 14:35:57 and 045 ms stored in a folder named by the current date, formatted as "`2024-04-16`".

c) When `take_photo.sh` is executed, a JSON metadata file with the same name as the photo but with the extension "`json`" must be saved in the same directory as the photo (also known as a sidecar file). The JSON file must contain the keys in this example:

```
{
      "File Name": "143567_045.jpg",
      "Create Date": "2024-04-16 14:35:57.045+02:00",
      "Create Seconds Epoch": 1713270957.045,
      "Trigger": "Time",
      "Subject Distance": 0.5574136009 m,
      "Exposure Time": "1/33",
      "ISO": 200
}
```

The key Trigger defines the cause of the capture which may have one of the following values:
`Time/Motion/External`

The values for the keys "Subject Distance", "Exposure Time" and "ISO" are available via the photos EXIF data which can for instance be read with the linux command `exiftool`

d) The wildlife camera must save a photo every 5 minutes with the key Trigger set to `Time`

e) The wildlife camera must take a photo approximately each second. If a motion is detected between any two consequitive photos, the latest photo must be saved with the key Trigger set to `Motion`

f) If an animal is detected by the external wildlife trigger (ESP8266), this information must immediately be received via MQTT. Based on this a photo must be taken with the JSON key Trigger set to External

g) If rain is detected by the Raspberry Pico (the BOOTSEL button is pressed) this info must be sent via MQTT to another script that then via MQTT requests a wipe of the lens screen (in a sequence of 0-180-0 degrees)

h) Via the wildlife camera access point a simple local website must provide access to all saved images and associated JSON metadata.

i) The wildlife camera must maintain a log functionality capturing all relevant events. The log file must be accessible via the simple local website.

j) The wildlife camera must become fully operational after a power failure without requiring user interaction (graceful shutdown can be assumed).

The functional requirements for the drone are:

k) When the drone is in the vicinity of the wildlife camera (i.e. the laptop can see the raspberry pi accesspoint SSID) the time of the wildlife camera must be synchronized with the time of the drone.
l) Also when the drone is in the vicinity of the wildlife camera, the drone must copy as many photos and associated metadata files not previously copied from the wildlife camera as possible. For each file succesfully copied the photo is retained on the wildlife camera. The copy must be registered on the wildlife camera by adding two keys to the metadata JSON file:

```
"Drone Copy": {"Drone ID": "WILDDRONE-001", "Seconds Epoch": 17132712340.458},
```

m) While connected to the wildlife camera, the drone must continuously log the WiFi link quality and signal level from /proc/net/wireless together with the seconds epoch. This should be logged to a simple sql database for subsequent use to optimize the flight path.

The functional requirements for the cloud are:

n) all photos must be annotated using ollama or a similar offline AI engine. The annotation text must be added to the metadata JSON file such as:

```
"Annotation": {"Source": "Ollama:7b", "Test": "The picture contains a lion cub
playing...."},
```

o) The metadata JSON files for a batch of offloaded photos with the added annotations must be committed to a folder within the team's git repository. Please notice that this is only the JSON files, not the photos themselves.

Nonfunctional requirements

a) The embedded system must be designed with the main tenets of the Unix philosopy in mind

b) Where appliable shell scipts should be used rather than other programming or scripting languages

c) Where applicable the embedded system must be scalable towards handling a large number of wildlife cameras (not near each other through).

d) Both the wildlife camera and and the drone (simulated by the linux desktop environment on a laptop computer) must to the extent possible be secured against malicious cyber attacks from both internet and the local wifi.

Main tenets of the Unix philosophy

Small is beautiful.

Make each program do one thing well.

Build a prototype as soon as possible.

Choose portability over efficiency.

Store numerical data in flat ASCII files.

Use software leverage to your advantage.

Use shell scripts to increase leverage and portability.

Avoid captive user interfaces.

Make every program a filter.

Rubrics

The rubrics below define the grading criteria for the individual presentation portfolio. The rubrics will be used as a qualitative measure, and the individual topics are not weighed equally.

Team project, document, video, git repository

	Excellent	Good	Needs improvement
Portfolio elements: Are the element listed in the description of the individual presentation portfolio present?	All required elements are present	One or two less critical elements are missing	Several or critical elements are missing
Project functional requirements: Are the functional requirements implemented and tested?	All functional requirements are implemented and tested	Most functional requirements are implemented and tested	Less than half of the functional requirements are implemented, tested and demonstrated
Project nonfunctional requirements: Does the solution design comply with the main tenets of the Unix philosopy. Are shell scripts used where applicable? Is the system design scalable towards handling multiple wildlife cameras? Is the RPi secured against malicious cyber attacks?	The solution design comply with the main tenets. Shell scripts are used where applicable, the system design is scalable towards handling multiple wildlife cameras. The RPi is well protected against malicious cyber attacks with multiple layers of security	The solution design comply mostly with the main tenets. Shell scripts are used for some applicable tasks, the system design is partly scalable towards handling multiple wildlife cameras. The RPi is reasonably protected against malicious cyber attacks with mostly a single layer of security.	The solution design comply only partly with the main tenets, few shell scripts are used, the system is not scalable towards handling multiple wildlife cameras. The RPi is not well protected against malicious cyber attacks.
Document organization and structure: How well-organized is the document? Is it following the requested structure? Are the main points presented in a logical order?	The document is well organized, following the requested structure and all main points are presented in a logical order.	The document is reasonably well organized, mostly following the requested structure, most main points are presented in a logical order.	The document is less well organized, not not fully following the requested structure, few main points are presented in a logical order.
Document writing style and visuals: Is the document content easily readable? Is the writing clear and concise? Does the report include relevant and effective visuals, such as images, graphs, tables, to help convey information?	The document is easily readable, relevant and effective visuals are used where applicable and are well explained	The document is readable, in most cases relevant visuals are used where applicable and are explained	The document is less readable, visuals are used to a limit extent, and are only partially explained

Are the visuals properly explained?			
Document content and accuracy: Does the document contain relevant and accurate information? Are sources properly cited? Is the information presented in a clear and understandable way?	The document contains relevant and accurate information with proper citations. The information is presented in a clear and understandable way	The document contains mostly relevant and accurate information with citations. The information is presented in an understandable way	The document contains less relevant and accurate information with few or no citations. The information is presented in a less understandable way
Solution approach: Are best practice approaches used for the overall architecture and design? Are advantages and drawback described?	Best practice approaches are used for the overall architecture and design. Advantages and drawbacks are well described.	Some best practice approaches are used for the overall architecture and design. Advantages and drawbacks are mostly described.	Few best practice approaches are used for the overall architecture and design. Advantages and drawbacks are vaguely or not described.
Solution description, test, results and conclusion: Are solutions to key tasks and concepts described, tested and concluded upon?	The solutions to key tasks and concepts are well described, tested and concluded upon.	Most solutions to key tasks and concepts are described, tested and concluded upon.	Few solutions to key tasks and concepts are described, tests are limited, and conclusion is superficial.
Video: Does the video clearly present and document the key outcomes of the project. Is the video easy to understand?	The video clearly presents and documents all key outcomes of the project. The video is easy to understand.	The video presents and documents most key outcomes of the project. The video is reasonable easy to understand	The video presents and documents few key outcomes of the project. The video is not easy to understand.
Git repository: Does the git repository provide a quick and intuitive access to the relevant material? Does it contain relevant scripts, configuration files, data file examples etc. created by the team for solving the project?	The git repository provides a quick and intuitive access to the relevant material. It contains all relevant scripts, configuration files, data file examples etc.	The git repository provides a reasonably quick and intuitive access to the relevant material. It contains most relevant scripts, configuration files, data file examples etc.	The git repository does not provide a quick and intuitive access to the relevant material. It contains only some relevant scripts, configuration files, data file examples etc.

Individual document

	Excellent	Good	Needs improvement
Organization and structure: How well-organized is the document? Is it following the requested structure? Are the main points presented in a logical order?	The document is well organized, following the requested structure and all main points are presented in a logical order.	The document is reasonably well organized, mostly following the requested structure, most main points are presented in a logical order.	The document is less well organized, not not fully following the requested structure, few main points are presented in a logical order.
Writing style and visuals: Is the document content easily readable? Is the writing clear and concise? Does the report include relevant and effective visuals, such as images, graphs, tables, to help convey information? Are the visuals properly explained?	The document is easily readable, relevant and effective visuals are used where applicable and are well explained	The document is readable, in most cases relevant visuals are used where applicable and are explained	The document is less readable, visuals are used to a limit extent, and are only partially explained
Content and accuracy: Does the document contain relevant and accurate information? Are sources properly cited? Is the information presented in a clear and understandable way?	The document contains relevant and accurate information with proper citations. The information is presented in a clear and understandable way	The document contains mostly relevant and accurate information with citations. The information is presented in an understandable way	The document contains less relevant and accurate information with few or no citations. The information is presented in a less understandable way
Discussion: Does the discussion compare the achieved solution, results and demonstration to the functional requirements? Is it discussed to what extent the solutions achieve the nonfunctional requirements?	The discussion presents an excellent comparison of the achieved solution, results and demonstration to the functional requirements. There is an excellent discussion of to what extent the solutions achieve the nonfunctional requirements. Excellent arguments are presented to substantiate this.	The discussion presents a good comparison of most of the achieved solution, results and demonstration to the functional requirements. There is a good discussion of to what extent the solutions achieve the nonfunctional requirements. Arguments are presented to substantiate this.	The discussion presents a limited comparison of most of the achieved solution, results and demonstration to the functional requirements. There is a limited discussion of to what extent the solutions achieve the nonfunctional requirements. Few arguments are presented to substantiate this.
Conclusion: Does the conclusion summarize	The conclusion summarizes well the	The conclusion summarizes most of the	The conclusion summarizes to a limited

the achieved solutions and outline how the solutions and results can be improved in future work?	achieved solutions and outlines well how the solutions and results can be improved in future work	achieved solutions and outlines how the solutions and results can be improved in future work	extent the achieved solutions and outlines to a limited extent how the solutions and results can be improved in future work
Learning goals: Does the student in the discussion and conclusion sections provide reflections based on knowledge and demonstrate application of the skills and competences defined in the learning goals?	The student provides excellent reflections based on learning goal knowledge and demonstrates excellent application of the learning goal skills and competences	The student provides reflections based on learning goal knowledge and demonstrates application of the learning goal skills and competences	The student provides few or no reflections based on learning goal knowledge and demonstrates limited application of the learning goal skills and competences