# IoT Project (5 ects) – Practical assignment outline

In the IoT project you will design, implement and report on an IoT smart system of your own. You can design the project to be whatever you like within the confines of the requirements presented in this document.

You can work in a group of 1-2 people.

The workload for design and implementation of the project is roughly 80 hours, with reporting and presenting consisting of another 40 hours.

The minimum required parts of a smart system are as follows:

* Raspberry Pi or Arduino which acts as the platform for the sensors
* 1-n sensors depending on the project
* A web service or server which stores up-to-date sensor data or other information, and provides some user interface

## SBC and sensors

You should use a single board computer (SBC, the Raspberry). You can also use an Arduino, or use both together (if you want). If using a Raspberry Pi use Python as the programming language – the development environment is up to you (you can use the Raspbian OS, or PL-App, or something else).

The SBC’s task is to control the sensors and send data to the web service.

There are various devices available in the classroom, as listed below. You can also use your own hardware.

* Sensor kits:
  + <https://www.sunfounder.com/learn/category/Sensor-Kit-v2-0-for-Arduino.html>
  + <https://www.instructables.com/id/Arduino-37-in-1-Sensors-Kit-Explained/>
  + <https://www.aliexpress.com/item/32932544732.html>
* RFID readers (there’s only 4 of these in class, though)
  + <https://www.aliexpress.com/item/32890412671.html>
  + <https://www.verkkokauppa.com/fi/product/69755/jnhdg/Raspberry-Pi-MFRC-522-13-56-MHz-RFID-lukija>
* Cameras for the Raspberry Pi
* PIR Motion sensors
* Many DHT22 temperature/humidity sensor modules
* some single items, like a soil moisture detector, and SD card reader for Arduino

You should start by familiarizing yourself with the sensors available, and then make a plan of what you can do with them.

## The web service

The web service must be hosted separately (that is, you cannot install a web server on the Pi itself).

The implementation of the web service is up to you – use any technologies you want or have picked up from programming courses. An example of how to do this would be to use Node.js and express to create a REST API.

## Examples of what you could do

The idea of the project should be self-invented. If not unique, you should have some reasonable real world motivation to develop the project. You could, among other things, create:

* Some application for traffic or environment monitoring
* An application for agriculture or a greenhouse
* An application for a smart home

## Planning

The first task is to plan the project. The plan can be informal, 1-2 pages. The plan should specify

* What system are you building?
* What components does it have?
* Who are the team members and what are their responsibilities (who does what).

Also: Specify the resources you will need to program / configure / build the project. That is, if you plan to use some hardware (e.g. a sensor), you must specify where you will get the manual for that device, how it is programmed and how you will be able to use it.

Plan how you will be using your time. Remember that there are 60 hours reserved for the design and implementation. Specify, how each member of the group will use that time (and be realistic!).

The plan needs to be submitted to Learn **latest by Wednesday November 4th**. The teacher will approve or ask for corrections by the end of that week. Your plan must be approved before starting to work with the project; unapproved projects will not be graded.

## Reporting

The report must use the reporting template. Formatting should be the same as in a thesis. The **deadline for the report is Sunday December 20th** .

The report should also use the same guidelines as the thesis to form an academic research report. The report must contain a background (theory) section and distinguish a theoretical framework in the scope of the work. You can use the example (research) reports given in Learn as examples of what you should be doing.

There is no formal page limit, but as a guideline aim for about 20 pages.

The report should consist of the following sections

* Introduction
* Background
* Implementation
* Discussion and conclusion

The theory in the report should be comprehensive. You should use at least an equal amount of scientific sources (Journal articles, conference proceedings, book chapters from scientific publishers) as other sources (usually web pages or other books). Usually at least 1.5 references per page is required in the theory section. Use at least 7 works as references.

The implementation section should present your projects technical implementation, and showcase how your prototype solution works. Describe both the hardware and software. Use appropriate diagrams to describe the architecture from both the hardware and software perspective. Also outline how the different components of the prototype communicate with each other. In practice, use at least

* A class diagram for describing the software components
* A message sequence diagram for describing the communication between components
* A flowchart describing the very high-level logic of you prototype

## Presentation

The project you have built must be presented either in class or in a Teams video call. The presentation is also informal; You should demonstrate to the teacher and your classmates what you have build and how it works. You must reserve a time for the presentation – this will be done according to instructions on Learn. The reservation is on first-come first-serve basis.

## Building and using school hardware

The devices in the classroom are for everyone’s use. Please courteous in their use, clean up after you are finished working etc.

Never take anything away from the classroom! There may be possibilities for borrowing equipment for a few days, but this must always be agreed with the teacher. If you want to work from home get your own equipment!