**Instructions for completing the course**

Start the course by looking at an **introduction to digital technology**. Pay special attention to the **sampling frequency** and its significance.

Next, you can start getting acquainted with digital **number systems**. Important in number systems are conversions from bits to hexadecimal numbers. The number of bits should be seen in its **magnitude at a glance**. The use of a floating point number and a fixed point number should be understood.

* Make one task of your choice on each assignments page that you return to Moodle. There will be a total of **seven assignments** in this section.

It should be realized **from the codes** that a bit frame can express various things related to application technology.

* Return one task from each assignments page to Moodle (you can choose). One of the task asks for a code to calculate parity check. Here is enough, the so-called pseudocode. The pseudocode shows how to do this, but the syntax of the programming language is ignored. In total, there will be **two tasks in this section**.

In combinatorial logic, the most important thing is to learn to think things logically. Circuits and minimization methods are automatic outputs when designed correctly.

* Combinational logic material returns at least two tasks from each task set. If there is only one task, that is enough. Here you need the LogiSim simulator. Install it on your machine. You simulate traffic light control with LogiSim. There will be a total of **nine assignments in this section**.

**State machine design** binds logic to real time. Often things happen at the clockwise. The basic idea is to learn how to design processes that work in tune with sensors signals or the clock. Again, the output (soft / hardware) is automatic if the state machine is basically functional.

* From this series of lectures, you restore the timeline of the Moodle **JK flip-flop** and the second **D-flip-flop**. In addition, three state machine tasks (you can choose which one you omit because there are four tasks).
* Familiarize yourself with the SW mode machine by watching the video “Software\_State Machine”. Design a state machine that keeps the robot with a black line. This continues as laboratory work
* There will be a **total of six tasks in this section**.

**The structure and operation of the CPU, the memory and testing** and the loading of data in the sections aim to create an insight into the current microcircuits and their operation.

* From these lectures, you return all assignments to Moodle. There will be a total of **three tasks in this section**.

**There are three laboratory jobs**. In the first work, the NAND circuits design the logic that keeps Autorobot on track. This can be accomplished when all tasks up to combinatorial logic have been completed and restored. Another job is to control the stepper motor with a state machine. Here you will also get familiar with the MSP430 processor. By the way, the third laboratory is the same as the first, but now the robot should be able to follow the track with the help of a state machine. The state machine is implemented in a processor environment. This lab can be done after all state machine tasks have been completed and returned. **In both labs, the plan must be returned for inspection before the laboratory.**

There will be a total of 24 tasks to be returned and two laboratories