Mock exam Embedded systems

Note: this exam is shorter than the real exam (this one is designed to last for 2.5 hours, the real one will last for 3.5 hours).

1. Explain clock stretching in the I^2C protocol.

2. We want to program a tyre-safety system. This system monitors the state of the four tyres of a car.

   The following tasks should be performed:

   - Measuring the radius of each tyre (this is done sequentially for each one): an ultrasound pulse is sent by applying a high voltage during 100 μs on the pin connected to the sensor. At least 1 ms later, the sensor responds by sending a pulse between 1 and 10 ms, whose duration is proportionnal to the tyre radius. Since this communication happens on a single pin, it is necessary to dynamically configure it as input or output. A pulse duration less that 5 ms is the symptom that tyre pressure is too low. In this case, the user has to be warned. The radius of each tyre should be monitored every second.
   - Measuring the stress inside the rubber (this is done sequentially for each one): a resistor is glued inside the rubber. Its resistance is a function of the mechanical stress of the rubber. The resistance is measured by reading the voltage across it with an analog-to-digital converter (ADC). A new conversion is launched by raising a flag, then the ADC signals that this conversion is completed with an interrupt. The conversion result is available in a dedicated register. A conversion takes at most 5 ms and this measurement has to be performed every second for each tyre. A voltage less than 2 V or greater than 4 V should trigger a warning.
   - A UART connection is responsible for sending information to the user on a screen. Every second, the four pressure values as well as the four stress values have to be sent. This is done by copying a few bytes to the registers of the UART peripheral and raising appropriate flags.
   - If a hazard is detected, a high voltage has to be immediately applied to a dedicated pin.

   Several timers are available, can be configured to various frequencies, and trigger interrupts.

   What is the best software architecture for this system ?

   Give a pseudo-code: precise enough to show tasks, communication mechanisms between them, and interactions with peripherals.

3. We want to model a dishwasher with a hybrid system.

   The dishwasher is composed of two controllers:

   - One for water level.
   - A second one for water temperature.

   The level controller works as follows: when the level is above 50 cm, it opens a drain valve until the level drops below 40 cm. When the valve is opened, the level decreases at a rate between 1 and 5 centimeter per second. We can assume that opening and closing of the valve are instantaneous. When the level is below 20 cm, a sprinkler adds water from the tank (provided that its temperature is above 50°C at that moment) when the level goes above 45 cm, it waits for a random time between 0 and 2 seconds and then stops. The filling rate is equal to 5 cm/s.

   Temperature evolution is described as follows: when the sprinkler is on, cold water from the tap is added to the tank, this decreses water temperature at a rate between 2 and 10 °C/min. When temperature is below 60 °C, the heater is turned on and increases it by 10°C/min. These figures are additive. The heater stays on until water temperature is greater or equal to 90°C.

   Initially, the level is zero, the valve is closed, the sprinkler is on, and water in the tank is boilling (100°C).

   Model this with a hybrid system.

   Give the three first steps of state-space exploration