Baking station – report

Yekta Buğrahan Nizamoğlu and Gyarmathy Tímea

The baking station was implemented using Finite State Machines for each of its components. The hardware is controlled from the software by taking into consideration the data from the sensors, the state of the FSMs and sending signals to the actuators accordingly.

Software

Each component has its own FSM defined, which goes through a series of states in order to simulate the flow of the baking station.

The mixer and the two turntables (Drehtisch) have a special initial state which check at the startup, if the components are in their desired position. If they are not, they adjust themselves by rotating, or in case of the mixer, it lifts up.

In order to solve the problem of being able to handle more boxes one after another, we wrote a simple FSM for the SLE light sensor at the beginning which increments a box counter only once when a box passes, since the sensor is on for the entire time a box passes and we only need the number to be incremented once (when it switches). We are counting the leaving boxes actually already at the first band which turns on if there are boxes in the system (boxCount > 0) and decrements the counter when a box leaves the band. For this we added a separate state MOVE_BOX_OUT_FB1 which just decrements the counter if the sensor at the end of the rolling band (SIF1) turned off and moves on to the next state. In this next state if the sensor SIF1 is on again (the second box arrived) we stop the band and move back into waiting for the turntable to be available. Else, if the box got on the first turntable, we check if there are more boxes in the system and if yes, we proceed to moving them in (go back to MOVE_IN_FB1), else we just send the first rolling band into the idle state (IDLE_FB1).

Whenever the system has to wait for a specific amount of time, we have counters to be incremented every time the main loop passes through that state, and we check if it reached the desired number which is *number of desired seconds* times *number of ticks per second* defined in the scheduler.

The preheat of the oven is started when the dough is done mixing, and the box waits under the mixer until the oven is preheated enough.

We get the temperatur by exponential smoothing (takes into consideration the previous value) and a simple approximative linear function we came up with by measuring some corresponding values. We use this temperature to control the PWM on the oven, but only within a 4 grade range below the desired temperature.

Hardware

We have 2 registers controlling the 5V actuators whose clock signal is given by the corresponding bus and the write signal, while for the 2 registers giving us the 5V sensor data, the clock is given by their bus and the read signal. These last ones should be available always but not when we write, so their output enable which is low-active is just connected to the negation of their clock.

For the 24V sensors and actuators we again have 1-1 register, except the actuators, respectively the sensors are decoupled from the 5V circuit with the help of a octocoupler each, which transforms 5V signals into 24V ones and for the sensors inversely. The actuators also use the help of a P-mos each, which has a voltage divider at its base in order to get only the allowed voltage there.

To read the get the temperature from the thermoresistor, we have an ADC connected between a voltage divider formed by this thermoresistor and a 2.8kOhm resistor, and register, and gives 8 bit data depending on the read value. Our circuit uses continuous conversion, which means we have to give a start signal so the ADC will work. Our start-up circuit is made out of a resistor and a capacitor. Until the capacitor fills up, the start signal gets 5V, then it is pulled to ground. The register's clock is given by the ADC when the output is ready.