Exercise 2 – Modelling & Implementation

16.10.2025, kuex

After modeling the washing machine control system in last week's exercise, it's time to bring things to action. From the washing machine supplier, you receive SW libraries that shall be used to control the washing machine. The functionality of the libraries and the HW schematics of the washing machine are available in Annex A and B, respectively.

Tasks:

- Read the annexes and check the available code to understand the code snippets the washing machine supplier provided.
- Define a modular structure of your software and define an API for all Inputs and Outputs for your washing machine.
- Implement the state machine according to your own specification from last week's exercise by extending the Keil uvision project available on Moodle
- Create mockup functions for all HW accesses implementing your defined API, so that the available buttons, dip switches and LEDs of the CT board can be used to emulate the washing machine.
- Describe, how you proceed to test your implementation. Can you guarantee that the washing machine works as expected? Does it work as described in last week's requirements?

Present your solution to the lecturer.

Annex A: SW Libraries

The module “timer” helps to implement timed behavior of the washing machine.

- You can start a timer calling timer_start(duration), where duration denotes the number of cycles of a 100Hz signal.
- You can stop a timer calling timer_stop().

The module “event_handler” helps you to detect events.

- Extend eh_get_event() such that you can recognize timer events. If the current value of timer equals 0, and the previous value of the timer has been different to 0, an Event TIME_OUT shall be generated

The module “action_handler” shall be programmed by you to implement the necessary actions. Please check the already existing code snippets.

- Extend the function’s prototypes with the implementation using the HW inputs and outputs as described in Appendix A and ennis.zhaw.ch.

The module “state_machine” shall contain your code implementing the state machine from last week’s lab.

- Extend the function “fsm_handle_event()” to implement your model’s behavior.

Annex B: HW schematics

HW

Figure 1 shows the schematics of all inputs and outputs of the washing machine.

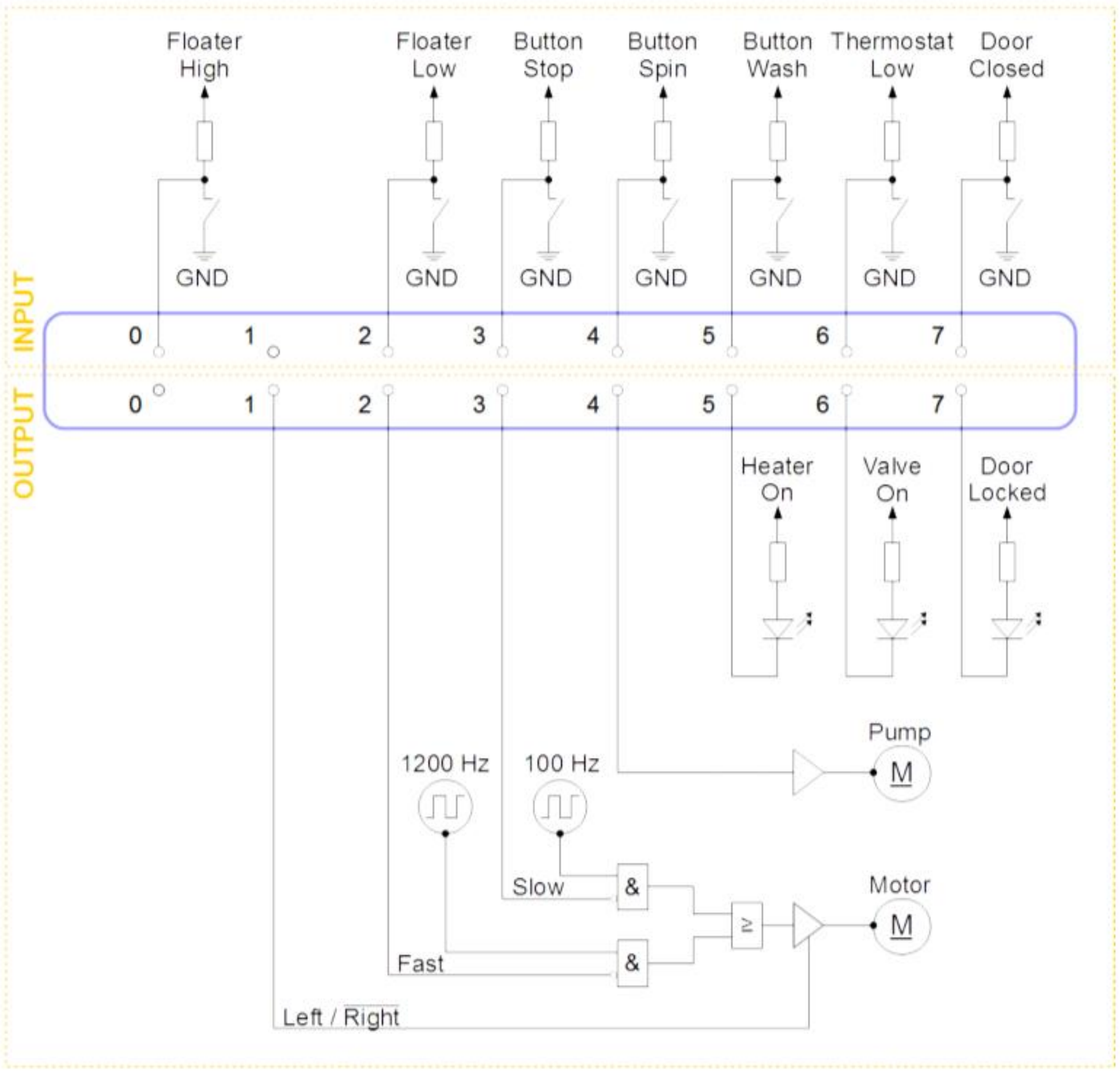

Figure 1: Inputs and Outputs

All inputs are "active-low", i.e. a logical "**0**" at the input "Door Closed" signals to the control unit that the door is closed. Edges indicate events. Hence, a falling edge on the signal "Door Closed" indicates that the door has been closed, a rising edge indicates the door's opening.

All outputs are "active-low", too. E.g., if the motor shall run at slow speed, output 3 shall be a "**0**" and consequently, output 2 shall be a "**1**".

Figure 2 shows the pin-out on the CT-board connector for Inputs and Outputs.

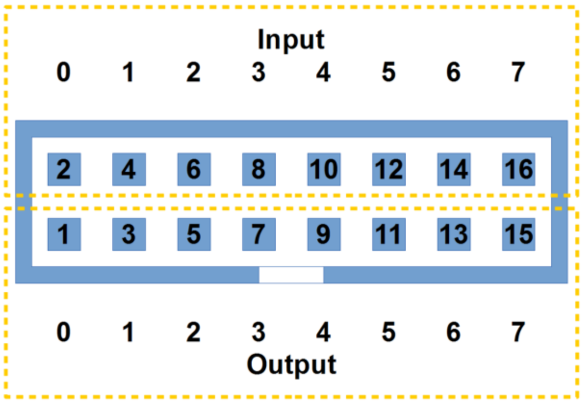

Figure 2: Pin-out for Input/Output connector on CT board

The connector can be accessed through the following memory addresses:

- Input: 0x60000410
- Output: 0x60000400

See also ennis.zhaw.ch for programming examples and further details:
https://ennis.zhaw.ch/wiki/doku.php?id=ctboard:peripherals:gpio_cpld