A remarkable machine part that we implemented was the push sensor. This sensor calibrates the sorter when the sorting machine is booted up, so that the sorting mechanism is initially down. First, the sorting mechanism is moved up until it reaches the sensor. When the sensor has detected the sorter, the sorter will descend until it’s at its lowest point. Since only the detection of a white disc will trigger the sorter to go up, the sorter will remain down, and the black discs, because they have the same colour as the conveyor belt and so the colour detector won’t be able to detect them.

Next, we use a timer as an input. When a certain action is performed, the timer is set to a specific value and starts to count down. That value was chosen by us; we measured each movement ten times and took the average. When the timer reaches zero, the machine transitions into a new state.

We chose to have the position sensor and colour detector to be constantly on. Since one of our priorities is that we’re focusing on the speed of the disc sorter, the feeder engine is always on during the sorting process. Consequently, the engine of the conveyor belt should always be on as well, in case a disc ends up on the conveyor belt. As both engines are always on, both the position sensor and colour detector are to be turned on, too.

As for our program that controls the PP2, we’ve had to design the code in Java first. This facilitates translating a working program to Assembly language. This Java program is translated to PHP code first, because we have a PHP to Assembly Language compiler from a previous course. As Java and PHP are quite similar, hardly anything needed to be changed, and thus it’s easy to write Assembly code.