LED AND BUTTONS

To manipulate the process during the computation, we use 2 buttons to start and output results. By pressing the start button, XMOS should start to reading and processing an input image and evolve for numerous times. To help the user understand the processing cycle, we use a green LED to indicate reading and flashing for processing inputs, however to prevent a deadlock or infinite looping happening during runtime, we also make an alternative green LED that reversing its state for each evolution of the game, thus the progression of the game can be visualized. To stop at a certain stage of evolution and produce the current result, we use another button to end the game, for exporting the result, we use blue LED to present the progression. Nevertheless, the user need to be able to pause the game at any given time during processing, this has already been done by physically tilting the board, and resume if placed horizontal again. But we also present this state with a red LED, and also displaying information of the current game on the console so the user can check the status of the game by pausing the process.

When we approach the idea of displaying information, we would like to keep the process as fast as possible, which means more workers and less channel. However, since LED and Button functions require to be visited constantly so it can check the input and output constantly, if they are separated, then that indicates each workers will need 2 channels connect to each functions instead of 1. Thus we made a decision of merging them into 1 function with case() that decides which function the data should be operated on, which mean the channel will be reduced, and the amount of workers will be increased.