Power Off

- **Actor** – The person or people who will perform the steps of this use case.

The person who pushes the button to power off the system.

- **Preconditions** – A description of the relevant and non-trivial state(s) of the system prior to the use case starting.

The System is on.

- **Normal course** – A description of the use case itself. This description can either be in narrative form, or a numbered list (1..N) of specific user steps. When a use case (such as “User approves/rejects customer requests”) has more than one way that a user can accomplish the needed steps, the most common way is shown here – only a single path is shown.

The system goes from on to off.

- **Alternate courses** – Descriptions of alternatives to, or deviations from the normal course. For example, the most common course might be to view the oldest unaddressed customer requests. An alternate course may be to view the unaddressed requests from the largest customers.

The player doesn’t turn off the machine. The machine loses power.

- **Exception courses** – Descriptions of what the user will experience when something goes wrong.

The user will experience system failure, or the machine will “freeze” and does not turn off.

- **Post-conditions** – Description of the affected portions of the state of the system after the use case has completed.

The system is now off.

- **Frequency of use** – An estimate of how often a particular use case will be exercised.

Every time the machine has been used by a player.

- **Assumptions** – Any assumptions that are implicit in the definition of the use case.

The hardware is not malfunctioning. The system is properly connected.

Power On

- **Actor** – The person or people who will perform the steps of this use case.

The person who pushes the button to start the system.

- **Preconditions** – A description of the relevant and non-trivial state(s) of the system prior to the use case starting.

The System is off.

- **Normal course** – A description of the use case itself. This description can either be in narrative form, or a numbered list (1..N) of specific user steps. When a use case (such as “User approves/rejects customer requests”) has more than one way that a user can accomplish the needed steps, the most common way is shown here – only a single path is shown.

The system goes from off to on and the default run number is set 1 and the channels are set to disarmed, also the default competition is IND.

- **Alternate courses** – Descriptions of alternatives to, or deviations from the normal course. For example, the most common course might be to view the oldest unaddressed customer requests. An alternate course may be to view the unaddressed requests from the largest customers.

The player doesn’t turn on the machine.

- **Exception courses** – Descriptions of what the user will experience when something goes wrong.

The user will experience nothing if the system fails to turn on.

- **Post-conditions** – Description of the affected portions of the state of the system after the use case has completed.

The system is now on.

- **Frequency of use** – An estimate of how often a particular use case will be exercised.

Every time the machine is used by a player.

- **Assumptions** – Any assumptions that are implicit in the definition of the use case.

The hardware is not malfunctioning. The system is properly connected.

System Reset

- **Actor** – The person or people who will perform the steps of this use case.

The person who pushes the button to reset the system.

- **Preconditions** – A description of the relevant and non-trivial state(s) of the system prior to the use case starting.

The System is on.

- **Normal course** – A description of the use case itself. This description can either be in narrative form, or a numbered list (1..N) of specific user steps. When a use case (such as “User approves/rejects customer requests”) has more than one way that a user can accomplish the needed steps, the most common way is shown here – only a single path is shown.

The system goes from on to off to back on and sets the system to the initial state described in On.

- **Alternate courses** – Descriptions of alternatives to, or deviations from the normal course. For example, the most common course might be to view the oldest unaddressed customer requests. An alternate course may be to view the unaddressed requests from the largest customers.

The player doesn’t reset the machine. The machine fails to reset just turning off.

- **Exception courses** – Descriptions of what the user will experience when something goes wrong.

Loss of power.

- **Post-conditions** – Description of the affected portions of the state of the system after the use case has completed.

The system is now returned to initial conditions and state is on.

- **Frequency of use** – An estimate of how often a particular use case will be exercised.

Whenever the user feels the system must be reset.

- **Assumptions** – Any assumptions that are implicit in the definition of the use case.

The hardware is not malfunctioning. The system is properly connected. The system is On.

Set Time

- **Actor** – The person or people who will perform the steps of this use case.

The person who is monitoring the system performs the command or action to set the value for time.

- **Preconditions** – A description of the relevant and non-trivial state(s) of the system prior to the use case starting.

The System is on and functioning properly.

- **Normal course** – A description of the use case itself. This description can either be in narrative form, or a numbered list (1..N) of specific user steps. When a use case (such as “User approves/rejects customer requests”) has more than one way that a user can accomplish the needed steps, the most common way is shown here – only a single path is shown.

The time on the Chronotimer will be set to a value inputted by the user.

- **Alternate courses** – Descriptions of alternatives to, or deviations from the normal course. For example, the most common course might be to view the oldest unaddressed customer requests. An alternate course may be to view the unaddressed requests from the largest customers.

The player doesn’t set a time.

- **Exception courses** – Descriptions of what the user will experience when something goes wrong.

The system doesn’t correctly set the variable for time and the time is unaffected.

- **Post-conditions** – Description of the affected portions of the state of the system after the use case has completed.

The system’s time is set to the user inputted value.

- **Frequency of use** – An estimate of how often a particular use case will be exercised.

Upon initializing the system.

- **Assumptions** – Any assumptions that are implicit in the definition of the use case.

The hardware is not malfunctioning. The system is properly connected.

Toggle State of Channel

- **Actor** – The person or people who will perform the steps of this use case.

The person who pushes the button to toggle state (enable/disable) of the channel.

- **Preconditions** – A description of the relevant and non-trivial state(s) of the system prior to the use case starting.

The system is on. To enable the channel, the channel must be disabled. To disable the channel, the channel must be enabled.

- **Normal course** – A description of the use case itself. This description can either be in narrative form, or a numbered list (1..N) of specific user steps. When a use case (such as “User approves/rejects customer requests”) has more than one way that a user can accomplish the needed steps, the most common way is shown here – only a single path is shown.

If the channel is enabled, it is now disabled. If the channel is disabled, it is now enabled.

- **Alternate courses** – Descriptions of alternatives to, or deviations from the normal course. For example, the most common course might be to view the oldest unaddressed customer requests. An alternate course may be to view the unaddressed requests from the largest customers.

Channel remains in current state.

- **Exception courses** – Descriptions of what the user will experience when something goes wrong.

The channel gets stuck in one state or a neutral state or is between states and enables and disables continuously.

- **Post-conditions** – Description of the affected portions of the state of the system after the use case has completed.

The channel has changed state.

- **Frequency of use** – An estimate of how often a particular use case will be exercised.

Every time the user wants to enable or disable a channel.

- **Assumptions** – Any assumptions that are implicit in the definition of the use case.

The hardware is not malfunctioning. The system is properly connected.

Connect Sensor

- **Actor** – Pusher
- **Preconditions** – Power is ON
- **Normal course** – A sensor of type{*eye, gate, pad*} is connected and assigned to (*channel*)
- **Alternate courses** –
- **Exception courses** –
- **Post-conditions** – New sensor is active and assigned to channel
- **Frequency of use** – >= 2/race
- **Assumptions** – The hardware functions properly

Disconnect Sensor

- **Actor** – Pusher
- **Preconditions** – At least one sensor is connected and assigned to a channel
- **Normal course** – The sensor is disconnected from previously-assigned channel
- **Alternate courses** –
- **Exception courses** –
- **Post-conditions** – One less sensor is connected, and one channel is now live but unassigned to a sensor
- **Frequency of use** – 4x/day of events
- **Assumptions** –

EVENT <TYPE>

- **Actor** – Pusher
- **Preconditions** – Power is ON, no event type has yet been assigned
- **Normal course** – User selects an event type from available options, in 1.0 options limited to <IND>
- **Alternate courses** –
- **Exception courses** –
- **Post-conditions** – Selected event type is saved and settings are configured properly according to stored info about the selected event type
- **Frequency of use** – 100x/race day
- **Assumptions** –

NEWRUN

- **Actor** – Pusher
- **Preconditions** – No race currently active, power ON, event type selected, 2 sensors active and assigned to channels
- **Normal course** – User initiates new run, START sensor fires, END sensor fires
- **Alternate courses** –
- **Exception courses** –
- **Post-conditions** – A run is in progress
- **Frequency of use** – 200x/race day
- **Assumptions** –

ENDRUN

- **Actor** – Pusher
- **Preconditions** – Race currently active
- **Normal course** –race ends and data is saved
- **Alternate courses** –
- **Exception courses** –
- **Post-conditions** – A set of run data issaved: time of day @ beginning of run, type of event, length of run (ie final time)
- **Frequency of use** – 200x/race day
- **Assumptions** – Everytime a race is initiated by the pusher, it represents an actual valid run for which a data set should be saved.

PRINT<RUN>

- **Actor** – Pusher
- **Preconditions** –Run currently active
- **Normal course** –user activates PRINT and the selected <RUN> is printed to the console via stdout
- **Alternate courses** –
- **Exception courses** –
- **Post-conditions** –Run data printed to console
- **Frequency of use** – 50x/race day
- **Assumptions** –

EXPORT<RUN>

- **Actor** – Pusher
- **Preconditions** –At least 1 run has been completed and saved
- **Normal course** –user activates EXPORT and the selected <RUN> is exported in XML to a file: “RUN<RUN>”
- **Alternate courses** –
- **Exception courses** – If the selected RUN has already been exported and the file currently exists in the export target directory, the system will not export a duplicate file
- **Post-conditions** –File “RUN<RUN>” is written
- **Frequency of use** –
- **Assumptions** –

NUM <NUM>Function

- **Actor** – The person or people who will perform the steps of this use case.

The person who triggers the function.

- **Preconditions** – A description of the relevant and non-trivial state(s) of the system prior to the use case starting.

The system is on and has power. Someone is currently waiting to be timed.

- **Normal course** – A description of the use case itself. This description can either be in narrative form, or a numbered list (1..N) of specific user steps. When a use case (such as “User approves/rejects customer requests”) has more than one way that a user can accomplish the needed steps, the most common way is shown here – only a single path is shown.

The person who is waiting to be timed is set at the number given by the actor to the next competitor to be timed.

- **Alternate courses** – Descriptions of alternatives to, or deviations from the normal course. For example, the most common course might be to view the oldest unaddressed customer requests. An alternate course may be to view the unaddressed requests from the largest customers.

The player doesn’t trigger the NUM function. The machine is powered off and no functions can be triggered.

- **Exception courses** – Descriptions of what the user will experience when something goes wrong.

The machine won’t set the athlete as the next competitor to start and the screen will display an error or the machine will freeze.

- **Post-conditions** – Description of the affected portions of the state of the system after the use case has completed.

The competitor is set at the given number to be the next competitor to be timed.

- **Frequency of use** – An estimate of how often a particular use case will be exercised.

Every time the machine is used to time a player.

- **Assumptions** – Any assumptions that are implicit in the definition of the use case.

The start button and finish button are connected to the right channel. The machine has power and the machine is not malfunctioning.

Clear<NUM>Function

- **Actor** – The person or people who will perform the steps of this use case.

The person who triggers the function.

- **Preconditions** – A description of the relevant and non-trivial state(s) of the system prior to the use case starting.

The system is on and has power. Someone is currently in the queue at the given number.

- **Normal course** – A description of the use case itself. This description can either be in narrative form, or a numbered list (1..N) of specific user steps. When a use case (such as “User approves/rejects customer requests”) has more than one way that a user can accomplish the needed steps, the most common way is shown here – only a single path is shown.

The actor triggers the clear function at the given number and the give number is removed from the queue.

- **Alternate courses** – Descriptions of alternatives to, or deviations from the normal course. For example, the most common course might be to view the oldest unaddressed customer requests. An alternate course may be to view the unaddressed requests from the largest customers.

The player doesn’t trigger the clear function. The machine is powered off and no functions can be triggered.

- **Exception courses** – Descriptions of what the user will experience when something goes wrong.

The machine won’t clear the given number from the queue and the screen will display an error or the machine will freeze.

- **Post-conditions** – Description of the affected portions of the state of the system after the use case has completed.

The give number is cleared from the queue.

- **Frequency of use** – An estimate of how often a particular use case will be exercised.

Every time the machine is used to time a player.

- **Assumptions** – Any assumptions that are implicit in the definition of the use case.

The start button and finish button are connected to the right channel. The machine has power and the machine is not malfunctioning.

SWAP<NUM>Function

- **Actor** – The person or people who will perform the steps of this use case.

The person who triggers the function.

- **Preconditions** – A description of the relevant and non-trivial state(s) of the system prior to the use case starting.

The system is on and has power. There are currently at least two people in the queue waiting to be timed.

- **Normal course** – A description of the use case itself. This description can either be in narrative form, or a numbered list (1..N) of specific user steps. When a use case (such as “User approves/rejects customer requests”) has more than one way that a user can accomplish the needed steps, the most common way is shown here – only a single path is shown.

The actor triggers the Swap function and the next two competitors in the queue are swapped.

- **Alternate courses** – Descriptions of alternatives to, or deviations from the normal course. For example, the most common course might be to view the oldest unaddressed customer requests. An alternate course may be to view the unaddressed requests from the largest customers.

The player doesn’t trigger the SWAP function. The machine is powered off and no functions can be triggered.

- **Exception courses** – Descriptions of what the user will experience when something goes wrong.

The machine won’t swap the first two competitors from the queue and the screen will display an error or the machine will freeze.

- **Post-conditions** – Description of the affected portions of the state of the system after the use case has completed.

The next two competitors in the queue are swapped.

- **Frequency of use** – An estimate of how often a particular use case will be exercised.

Every time the machine is used to time a player.

- **Assumptions** – Any assumptions that are implicit in the definition of the use case.

The start button and finish button are connected to the right channel. The machine has power and the machine is not malfunctioning.

DNF Function

- **Actor** – The person or people who will perform the steps of this use case.

The person who triggers the channel.

- **Preconditions** – A description of the relevant and non-trivial state(s) of the system prior to the use case starting.

The system is on and has power. Someone is currently still being timed.

- **Normal course** – A description of the use case itself. This description can either be in narrative form, or a numbered list (1..N) of specific user steps. When a use case (such as “User approves/rejects customer requests”) has more than one way that a user can accomplish the needed steps, the most common way is shown here – only a single path is shown.

The person who is still being time will not finish the race and their time in the system will stop being timed.

- **Alternate courses** – Descriptions of alternatives to, or deviations from the normal course. For example, the most common course might be to view the oldest unaddressed customer requests. An alternate course may be to view the unaddressed requests from the largest customers.

The player doesn’t trigger the DNF function. The machine is powered off and no channels can are triggered.

- **Exception courses** – Descriptions of what the user will experience when something goes wrong.

The machine won’t stopcounting and the screen will display an error or the machine will freeze.

- **Post-conditions** – Description of the affected portions of the state of the system after the use case has completed.

The timer has stopped recording the time for that player.

- **Frequency of use** – An estimate of how often a particular use case will be exercised.

Every time the machine is used to time a player.

- **Assumptions** – Any assumptions that are implicit in the definition of the use case.

The start button and finish button are connected to the right channel. The machine has power and the machine is not malfunctioning.

Trigger

- **Actor** – The person or people who will perform the steps of this use case.

The person who triggers the channel.

- **Preconditions** – A description of the relevant and non-trivial state(s) of the system prior to the use case starting.

The system is on and has power.

- **Normal course** – A description of the use case itself. This description can either be in narrative form, or a numbered list (1..N) of specific user steps. When a use case (such as “User approves/rejects customer requests”) has more than one way that a user can accomplish the needed steps, the most common way is shown here – only a single path is shown.

The person triggers a channel which will either start or stop the channel depending on which is triggered.

- **Alternate courses** – Descriptions of alternatives to, or deviations from the normal course. For example, the most common course might be to view the oldest unaddressed customer requests. An alternate course may be to view the unaddressed requests from the largest customers.

The player doesn’t trigger the channel. The machine is powered off and no channels can be triggered.

- **Exception courses** – Descriptions of what the user will experience when something goes wrong.

The machine won’t stop or start counting and the screen will display an error or the machine will freeze.

- **Post-conditions** – Description of the affected portions of the state of the system after the use case has completed.

The timer has either started or stopped

- **Frequency of use** – An estimate of how often a particular use case will be exercised.

Every time the machine is used to time a player.

- **Assumptions** – Any assumptions that are implicit in the definition of the use case.

The start button and finish button are connected to the right channel. The machine has power and the machine is not malfunctioning.

Start Trigger

- **Actor** – The person or people who will perform the steps of this use case.

The person who pushes the button to start the timer or the athlete who triggers the sensor to start the timer.

- **Preconditions** – A description of the relevant and non-trivial state(s) of the system prior to the use case starting.

The timer is not running. The system is on.

- **Normal course** – A description of the use case itself. This description can either be in narrative form, or a numbered list (1..N) of specific user steps. When a use case (such as “User approves/rejects customer requests”) has more than one way that a user can accomplish the needed steps, the most common way is shown here – only a single path is shown.

The timer starts counting and recording the time.

- **Alternate courses** – Descriptions of alternatives to, or deviations from the normal course. For example, the most common course might be to view the oldest unaddressed customer requests. An alternate course may be to view the unaddressed requests from the largest customers.

The player doesn’t start the timer. The timeris already started and cannot start again. The machine loses power.

- **Exception courses** – Descriptions of what the user will experience when something goes wrong.

The machine won’t start counting and the screen will display an error or the machine will freeze.

- **Post-conditions** – Description of the affected portions of the state of the system after the use case has completed.

The timer has now started counting. The timer is now recording the time.

- **Frequency of use** – An estimate of how often a particular use case will be exercised.

Every time the machine is used to time a player.

- **Assumptions** – Any assumptions that are implicit in the definition of the use case.

The start button is connected to the right channel. The machine has power and the machine is not malfunctioning.

Finish Trigger

- **Actor** – The person or people who will perform the steps of this use case.

The person who pushes the button to stop the timer or the athlete who triggers the sensor to stop the timer.

- **Preconditions** – A description of the relevant and non-trivial state(s) of the system prior to the use case starting.

The timer is running. The system is on.

- **Normal course** – A description of the use case itself. This description can either be in narrative form, or a numbered list (1..N) of specific user steps. When a use case (such as “User approves/rejects customer requests”) has more than one way that a user can accomplish the needed steps, the most common way is shown here – only a single path is shown.

The timer stops counting and records the time.

- **Alternate courses** – Descriptions of alternatives to, or deviations from the normal course. For example, the most common course might be to view the oldest unaddressed customer requests. An alternate course may be to view the unaddressed requests from the largest customers.

The player doesn’t stop the timer. The machine loses power.

- **Exception courses** – Descriptions of what the user will experience when something goes wrong.

The machine will not stop counting and the screen will display an error or the machine will freeze.

- **Post-conditions** – Description of the affected portions of the state of the system after the use case has completed.

The timer has now stopped. The time has now been recorded.

- **Frequency of use** – An estimate of how often a particular use case will be exercised.

Every time the machine is used to time a player.

- **Assumptions** – Any assumptions that are implicit in the definition of the use case.

The finish button is connected to the right channel. The machine has power and the machine is not malfunctioning.