

# A Sample Document for the Usages of **lstEventB** Package

Thai Son Hoang  
ECS, University of Southampton  
<T dot S dot Hoang at ecs dot soton dot ac dot uk>

November 4, 2018

For convenient, we define macro `\EventB` for Event-B.

We start first with some inline Event-B code by embedding them using a pair of `|`, for example `|@grd1: "SNSR = FALSE"|` gives `@grd1:"SNSR = FALSE"`. Any Event-B formulae including Unicode symbols will be typeset using the `bsymb` package accordingly.

ASCII	Symbols	Explanation
<code>!:</code>	$\in$	Set membership
<code>/:</code>	$\notin$	Set non-membership
<code>&lt;:</code>	$\subseteq$	Subset
<code>/&lt;:</code>	$\not\subseteq$	Not a subset
<code>&lt;&lt;:</code>	$\subset$	Proper subset
<code>/&lt;&lt;:</code>	$\not\subset$	Not a proper subset
<code>!finite</code>	finite	Finite
<code>!partition</code>	partition	Partition

Table 1: Set predicates

ASCII	Symbols	Explanation
<code>!BOOL</code>	BOOL	BOOL set
<code>!TRUE</code>	TRUE	TRUE
<code>!FALSE</code>	FALSE	FALSE
<code>!bool</code>	bool	bool predicate

Table 2: BOOL and bool

More complete piece of code (including the Unicode symbols) can be typeset using the `EventBcode` environment. Below is the typesetting of an Event-B machine.

---

ASCII	Symbols	Explanation
<code>!INT</code>	$\mathbb{Z}$	Set of integer numbers
<code>!NAT</code>	$\mathbb{N}$	Set of natural numbers
<code>!NAT1</code>	$\mathbb{N}_1$	Set of positive natural numbers
<code>!min</code>	min	Minimum
<code>!max</code>	max	Maximum
<code>-</code>	$-$	Difference
<code>*</code>	$*$	Product
<code>!/</code>	$\div$	Quotient
<code>!mod</code>	mod	Remainder
<code>..</code>	$..$	Interval

Table 3: Numbers

ASCII	Symbols	Explanation
<code>&gt;</code>	$>$	Greater
<code>&lt;</code>	$<$	Less
<code>&gt;=</code>	$\geq$	Greater or equal
<code>&lt;=</code>	$\leq$	Less or equal

Table 4: Number predicates

ASCII	Symbols	Explanation
<code>&lt;-&gt;</code>	$\leftrightarrow$	Relations
<code>!dom</code>	dom	Domain
<code>!ran</code>	ran	Range
<code>&lt;&lt;-&gt;</code>	$\Leftrightarrow$	Total relations
<code>&lt;-&gt;&gt;</code>	$\Rrightarrow$	Surjective relations
<code>&lt;&lt;-&gt;&gt;</code>	$\Leftrightarrow$	Total surjective relations
<code>!circ</code>	$\circ$	Backward composition
<code>!id</code>	id	Identity
<code>&lt; </code>	$\triangleleft$	Domain restriction
<code>&lt;&lt; </code>	$\triangleleft$	Domain subtraction
<code> &gt;</code>	$\triangleright$	Range restriction
<code> &gt;&gt;</code>	$\triangleright$	Range subtraction
<code>~</code>	$-1$	Inverse
<code>&lt;+</code>	$\Leftarrow$	Overriding
<code>&gt;&lt;</code>	$\otimes$	Direct product
<code>  </code>	$\parallel$	Parallel product
<code>!prj1</code>	prj <sub>1</sub>	First projection
<code>!prj2</code>	prj <sub>2</sub>	Second projection

Table 5: Relations

ASCII	Symbols	Explanation
$\leftrightarrow$	$\leftrightarrow$	Partial functions
$\rightarrow$	$\rightarrow$	Total functions
$\rightarrowtail$	$\rightarrowtail$	Partial injections
$\hookrightarrow$	$\hookrightarrow$	Total injections
$\twoheadrightarrow$	$\twoheadrightarrow$	Partial surjections
$\twoheadrightarrow$	$\twoheadrightarrow$	Total surjections
$\rightarrowtail$	$\rightarrowtail$	Bijections
$\lambda$	$\lambda$	Lambda abstraction

Table 6: Functions

ASCII	Symbols	Explanation
$:=$	$:=$	Becomes equal to
$:$	$:\in$	Choice from a set
$: $	$: $	Choice by predicate

Table 7: Functions

```

1 machine Sensor_m0_SNSR
2 variables
3 SNSR
4 invariants
5 @thm0_1: "SNSR ∈ BOOL" theorem
6 events
7
8 INITIALISATION
9 begin
10   @act1: "SNSR := FALSE"
11 end
12
13 SNSR_on
14 when
15   @grd1: "SNSR = FALSE"
16 then
17   @act1: "SNSR := TRUE"
18 end
19
20 SNSR_off
21 when
22   @grd1: "SNSR = TRUE"
23 then
24   @act1: "SNSR := FALSE"
25 end
26
27 end

```

---

One can change the different colour options. For example, `\EventBSetKeywordColour{blue!50!black}` will change the keyword colour to dark blue. (This has effects only when

---

```

1 machine Sensor_m0_SNSR
2 variables

```

```

3 SNSR
4 invariants
5 @thm0_1: "SNSR ∈ BOOL" theorem

```

---

One can include external file containing Event-B code using the `\EventBinutlisting` command. For example the following is the result of including the code in the file `Sensor_m1_DEP.bumx` using `\EventBinutlisting{Sensor_m1_DEP.bumx}`.

```

1 machine Sensor_m1_DEP
2 refines Sensor_m0_SNSR
3 variables
4 SNSR
5 DEP
6 invariants
7 @inv0_1: "DEP ∈ ℕ"
8 events
9
10 INITIALISATION extended
11 begin
12 @act2: "DEP := 0"
13 end
14
15 SNSR_on extended
16 refines SNSR_on
17 end
18
19 SNSR_off extended
20 refines SNSR_off
21 begin
22 @act2: "DEP := DEP + 1"
23 end
24
25 end

```

---

More specifically, one can specify more details on the inclusion, e.g., the ranges, as the following example `\EventBinutlisting[firstline=16,lastline=20]{Sensor_m2_snsr.bumx}` gives

```

1 @inv1_1: "Snsr_01 = TRUE ⇒ SNSR = TRUE"
2
3 @inv1_2: "Snsr_10 = TRUE ⇒ SNSR = FALSE"
4
5 @inv1_3: "Snsr_01 = FALSE ∨ Snsr_10 = FALSE"

```

---

```

1 machine Sensor_m3_Ctrl
2
3 refines
4
5 Sensor_m2_Snsr

```

---

```

6
7 variables
8
9   SNSR
10
11  DEP
12
13  Snsr_01
14
15  Snsr_10
16
17  ctrl_snsr
18
19  ctrl_dep
20
21  ctrl_snsr_01
22
23  ctrl_snsr_10
24
25 invariants
26
27  @inv2_1:
28  "Snsr_01 = FALSE  $\wedge$  Snsr_10 = FALSE  $\wedge$  ctrl_snsr_01 = FALSE  $\wedge$  ctrl_snsr_10 = FALSE
     $\Rightarrow$  ctrl_snsr = SNSR"
29
30  @inv2_2: "ctrl_dep  $\in \mathbb{N}$ "
31
32  @inv2_3: "Snsr_10 = FALSE  $\wedge$  ctrl_snsr_10 = FALSE  $\Rightarrow$  ctrl_dep = DEP"
33
34  @inv2_4: "Snsr_10 = TRUE  $\vee$  ctrl_snsr_10 = TRUE  $\Rightarrow$  ctrl_dep = DEP - 1"
35
36  @inv2_5: "ctrl_snsr_01 = TRUE  $\Rightarrow$  SNSR = TRUE"
37
38  @inv2_6: "ctrl_snsr_10 = TRUE  $\Rightarrow$  SNSR = FALSE"
39
40  @inv2_7: "ctrl_snsr_01 = TRUE  $\Rightarrow$  Snsr_01 = FALSE"
41
42  @inv2_8: "ctrl_snsr_10 = TRUE  $\Rightarrow$  Snsr_10 = FALSE"
43
44 events
45
46  INITIALISATION extended
47  refines INITIALISATION
48  begin
49    @act5: "ctrl_snsr := FALSE"
50    @act6: "ctrl_dep := 0"
51    @act7: "ctrl_snsr_01 := FALSE"
52    @act8: "ctrl_snsr_10 := FALSE"
53  end
54
55  SNSR_on extended
56  refines SNSR_on
57  when
58    @grd3: "ctrl_snsr_10 = FALSE"
59  end
60

```

```

61 SNSR_off extended
62 refines SNSR_off
63 when
64   @grd3: "ctrl_snsr_01 = FALSE"
65 end
66
67 ctrl_Senses_Snsr_01 extended
68 refines ctrl_Senses_Snsr_01
69 begin
70   @act2: "ctrl_snsr_01 := TRUE"
71 end
72
73 ctrl_Senses_Snsr_10 extended
74 refines ctrl_Senses_Snsr_10
75 begin
76   @act2: "ctrl_snsr_10 := TRUE"
77 end
78
79 ctrl_on
80 when
81   @grd1: "ctrl_snsr_01 = TRUE"
82 then
83   @act1: "ctrl_snsr_01 := FALSE"
84   @act2: "ctrl_snsr := TRUE"
85 end
86
87 ctrl_off
88 when
89   @grd1: "ctrl_snsr_10 = TRUE"
90 then
91   @act1: "ctrl_snsr_10 := FALSE"
92   @act2: "ctrl_snsr := FALSE"
93   @act3: "ctrl_dep := ctrl_dep + 1"
94 end
95
96 end

```