

# **CLOCKWORK LANGUAGE**

A look at the code

## **LANGUAGE FEATURES**

Rule based

Automatic state selection

No Loop statements

Highly reusable code

## DEFINING STATES

```
Light MACHINE {  
    on STATE;  
    off INITIAL;  
}
```

Simple machine

No automation

Controlled externally

## USING TRANSITIONS

```
ToggleSwitch MACHINE {  
    on STATE;  
    off INITIAL;  
  
    TRANSITION on TO off USING next;  
    TRANSITION off TO on USING next;  
}
```

Basic automation via transitions

Flips state when 'next' message is received

## SITUATION NORMAL

MACHINEs can have STATEs

TRANSITIONS can cause STATEs to change

Nothing special so far...

## USING RULES

```
Blinker MACHINE {  
    on WHEN SELF IS off;  
    off DEFAULT;  
}
```

Automatic state selection

Evaluates rules in sequence

Stops at the first rule that is true

# CONTROLLING THINGS

```
LightSwitch MACHINE switch, light {  
    on WHEN switch IS on;  
    off DEFAULT;  
  
    ENTER on { SET light TO on; }  
    ENTER off { SET light TO off; }  
}
```

Controls 'light' based on the state of 'switch'

# NO LOOPS OR IFS?

```
Counter MACHINE {  
    OPTION count 0;  
    up WHEN count < 10;  
    idle DEFAULT;  
  
    ENTER up { count := count + 1; }  
}
```

## NO IF STATEMENTS?

```
Counter MACHINE {  
  OPTION count 0;  
  up WHEN count < 10;  
  idle DEFAULT;  
  
  ENTER up {  
    count := count + 1;  
    IF count < 10 { SET SELF TO idle; }  
  }  
}
```

An ugly fix to the error in the rules

## COUNTING UP

```
Counter MACHINE {  
  OPTION count 0;  
  up WHEN SELF IS idle AND count < 10;  
  idle DEFAULT;  
  
  ENTER up { count := count + 1; }  
}
```

## REUSING COMPONENTS

```
LightSwitch MACHINE switch, light {  
    on WHEN switch IS on;  
    off DEFAULT;  
  
    ENTER on { SET light TO on; }  
    ENTER off { SET light TO off; }  
}
```

## CONTROLLING MULTIPLE THINGS AT ONCE

```
LightController MACHINE lights {  
    COMMAND on { SEND turnOn TO lights; }  
    COMMAND off { SEND turnOff TO lights; }  
}  
front_porch Light;  
garden Light;  
yard_lights LIST front_porch, garden;  
  
controller LightController yard_lights;
```

**LINUX.CONF.AU**  
21-25 January 2019 | Christchurch | NZ

**The Linux of Things**  
#LCA2019@linuxconfa