Une : a Declarative DSL for the Mingle Platform

MISCELLANEOUS

- Language is case **in**sensitive
- Command separator: LF + LF (blank line)
- Comments: # (ignores to the end of the line)
- \ continues in next line
- Commands can be in any order[1]
- Compiler always use UTF-8 for output

(1) The same for the clauses inside a command

DATA TYPES

BASIC

- BOOLEAN: { TRUE | FALSE }
  (Alias: CLOSED, ON, YES | OPEN, OFF, NO)
- NUMBER : Decimal (signed 32-bits: 1234.56 or 1_234.56)
- STRING : Everything in between double quotes (“ ”)

EXTENDED

date(), time(), list(), pair()

URI SCHEMES (between quotes)

- file:// (wildcards allowed)
- http:// (wildcards not allowed)
- https:// (wildcards not allowed)

NUMBERS SUFFIXES

Time : { u | t | s | m | h | d } (converts to millis)

Temperature: { c | f | k } (converts to Celsius)

OPERATORS

- Arithmetic: +, -, *, /, %, ^
  (Sum, Diff, Times, Divide,Mod, Power)
- Relational: ==, >, <, >=, <=, !=, <>
  (Alias: IS or EQUALS or ARE, BELOW, ABOVE, LEAST, MOST, UNEQUAL, IS_NOT)
- Conditional: &&, ||, |&, !
  (Alias: AND, OR, XOR, NOT)
- Bitwise: &, |, ><, ~ >> <<
  (Alias: BAND, BOR, BXOR, BNOT) (shifts)
- Assign: = (Alias: SET)
- Send (get/set - functions invocation) :

ENUNCIATIONS

<name[1]> : Letter, Digit[2] or ‘_‘ (max 48 chars)

<type> : Either a ‘basic’ or a ‘extended’ data type

<expression>: { <name> | <type> } <operator> <expression>

<literal> : <name> | <type> | <expression>

(1) In *Une*, ‘name’ is the same as ‘identifier’ or ‘label’ in other languages
(2) Names can not start with a digit

MACROS

Folders

- {*home*} MSP home
- {*home.inc*} includes
- {*home.lib*} libraries
- {*home.log*} logs
- {*home.tmp*} temporal

PREPROCESSOR

INCLUDE {“<URI [*|**](1)>” | “*”} [; …]

USE <name(s)> AS <literal> [; …]

(1) While * Includes all files in folder, ** includes also subfolders.
Use of asterisks works only with local files

SCRIPTS (for advanced users)

SCRIPT [<name>]
LANGUAGE { java | js | python | ...}
FROM { “<URI>” [; ...] | {<code>} }
[CALL “<entry_point>”]
[ONSTART] [ONSTOP]

DRIVERS (for advanced users)

DRIVER <name>
SCRIPT <script-name>
[CONFIG <prop> AS {ANY | <data_type>} [REQUIRED] [; ...]]

DEVICES (SENSORS & ACTUATORS)

DEVICE(1) <name>
[INIT <(2)prop> = {<literal> | <expresion(3)>} [; …]]
DRIVER <driver>
[CONFIG name = {<literal> | <expression(3)>} [; …]]

Example:

```
DEVICE LightKitchen
    INIT
        groups  SET “lights, ground_floor”
    (3) value   SET “an string”.len() ABOVE 0
    DRIVER driverForLights
        CONFIG
            IP SET ”192.168.1.110”
```

(1) SENSOR and ACTUATOR are synonyms (alias) of DEVICE
(2) In MSP is one of following: ‘groups’, ‘value’, ‘delta’, ‘downtime’
(3) If it is an expression, it must be resolvable at compilation time

RULES

[RULE <name>]
WHEN <precondition> [{AND | OR} […]]
THEN <actions> [; …]
[IF <future-condition] [{AND | OR} […]]
[USE <alias>] [; …]

PreConditions
[[ANY | ALL] <group_name>] <expression>

Actions
{<script> | <rule> | {<device> | <group>} = <expression>}
[AFTER <time_unit>]

Future-conditions (optional)
[[ANY | ALL] <group_name>] <expression>
[{AFTER | WITHIN} <time>]

Aliases (optional)
USE <name> AS <expression>

Examples:

```
WHEN Inside > 0K        # Zero Kelvin: true for every new value
    THEN console = “The temperature inside is “+ Inside +“ ºC”

WHEN alarm IS ON
    THEN all_lights SET OFF

WHEN ANY contact IS OPEN AND alarm IS ON
    THEN MyTelegram SET “Danger: intruders at home”
     IF (alarm IS ON) AFTER 20s

WHEN ANY door IS open OR ANY window IS open
    THEN heater SET off
    IF (ANY door IS open OR ANY window IS open) WITHIN 3m
```

RESERVED KEYWORDS

AFTER	ALL	ANY	AS
CALL	CONFIG	DEVICE	DRIVER
FROM	IF	INCLUDE	INIT
LANGUAGE	ONSTART	ONSTOP	REQUIRED
RULE	SCRIPT	THEN	USE
WHEN	WITHIN		For future →
ALIAS	BY	FOR	ONERROR

How to learn *Une*

Read “Une_language.pdf” and try the examples in the “examples” folder (they are ordered by difficulty) using the IDE (named “Glue”).

Visit: https://mingle.peyrona.com