Footprint_

string data
int num_table
float x1
float y1
float x2
float y2
Int status
void process_data()
void print(int n)
int action(int action)
void reset()

Data: string data for one footprint step.
Ex: data: ~x1=-0.41~~y1=-0.345~~x2=0.41~~y2=0.345~

num_table: just display for developer if they have run in terminal.

process_data(): get string store in data and process to have value and update to other variable.
Ex: data:data:~x1=-0.41~~y1=-0.345~~x2=0.41~~y2=0.345~. -> set x1=-0.41,y1=-0.345, x2=0.41, y2=0.345.

Footprint is defined by rectangle. There are two point to draw. In this case we use (x1,y1) and (x2,y2) to draw footprint. x1,y1, alway negative and x2,y2 is alway positive.

status : variable about status but not use in this case.

print(): just print all variable currently store in footprint step

action(): set new footprint for robot

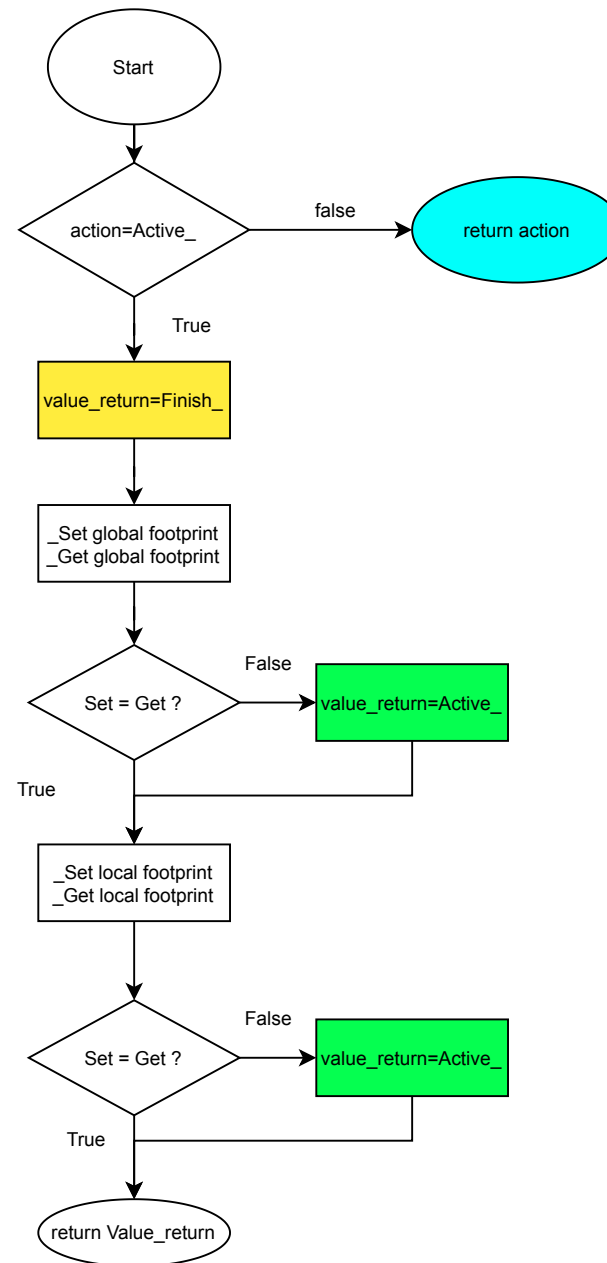reset(): nothing to reset. It is void function.

```
#define Error_  3
#define Active_ 1
#define Cancel_ 4
#define Finish_ 2
#define N_A_ 0
#define Skip_ 5
#define True_ 6
#define False_ 7
#define Wake_up_ 8
#define Stop_ 9
#define Break_ 10
#define is_robot 1
```

int action(int action)

```
                    ┌─────────┐
                    │  Start  │
                    └─────────┘
                         │
                         ▼
                    ◇ action=Active_ ◇ ──false──▶ ( return action )
                         │
                         │ True
                         ▼
              ┌──────────────────────┐
              │  value_return=Finish_ │
              └──────────────────────┘
                         │
                         ▼
              ┌──────────────────────┐
              │ _Set global footprint│
              │ _Get global footprint│
              └──────────────────────┘
                         │
                         ▼
                   ◇ Set = Get ? ◇ ──False──▶ [ value_return=Active_ ]
                         │                              │
                         │ True                         │
                         ▼◀─────────────────────────────┘
              ┌──────────────────────┐
              │ _Set local footprint │
              │ _Get local footprint │
              └──────────────────────┘
                         │
                         ▼
                   ◇ Set = Get ? ◇ ──False──▶ [ value_return=Active_ ]
                         │                              │
                         │ True                         │
                         ▼◀─────────────────────────────┘
                ( return Value_return )
```

when operating, call function "int acition(int action)" to operating. Created loop and set timer to call action. If step finish -> they are return Finish