Для полива системы

MST - влажность почвы
CRT - идеальная / совпадает с заданной
WTR - вода / полив

#define NO_CONNECTION_STATE 0
#define WAITING_STATE 1
#define SENSOR_ANALISYS_STATE 2
#define CHECK_WATER_STATE 3
#define WATERING_ON_STATE 4
#define ERROR_STATE 5

#define CONNECT_CMD '0'
#define GET_MST_CMD '1'
#define WTR_ON_CMD '2'
#define WTR_OFF_CMD '3'
#define WAT_LEV_OK_CMD '4'
#define WAT_LEV_BAD_CMD '5 '
#define WAIT_WTR_CMD '6'
#define CM DIS'D'_CONNECT 7'

Глобальные переменные:
int state - хранит в себе реальные показатели датчиков

void loop()

int cmd = get_command();

state

когда недостаточно воды в баке

ERROR_STATE
NO_CONNECTION_STATE
WAITING_STATE
SENSOR_ANALISYS_STATE
CHECK_WATER_STATE
WATERING_ON_STATE

errorSt_handler()
noConSt_handler()
waitSt_handler(cmd)
senAnSt_handler(cmd)
chWtSt_handler(cmd)
wtOnSt_handler(cmd)

void loop()

Watering userWatering

прописываем подкл к mesh и считываем запрос на смену времени полива

int get_command();

int cmd = (int_topic_time)

```
waterringTimeoutInSec = in_topic_time;
```

```
waterringTimeoutInSec = 10;
```

return;

uint32_t waterring_start_time; // Глобальная переменная хранит время запуска насоса

void chWtSt_handler()

bool waterLevel = checkWaterLevel();

waterLevel

state = ERROR_STATE;

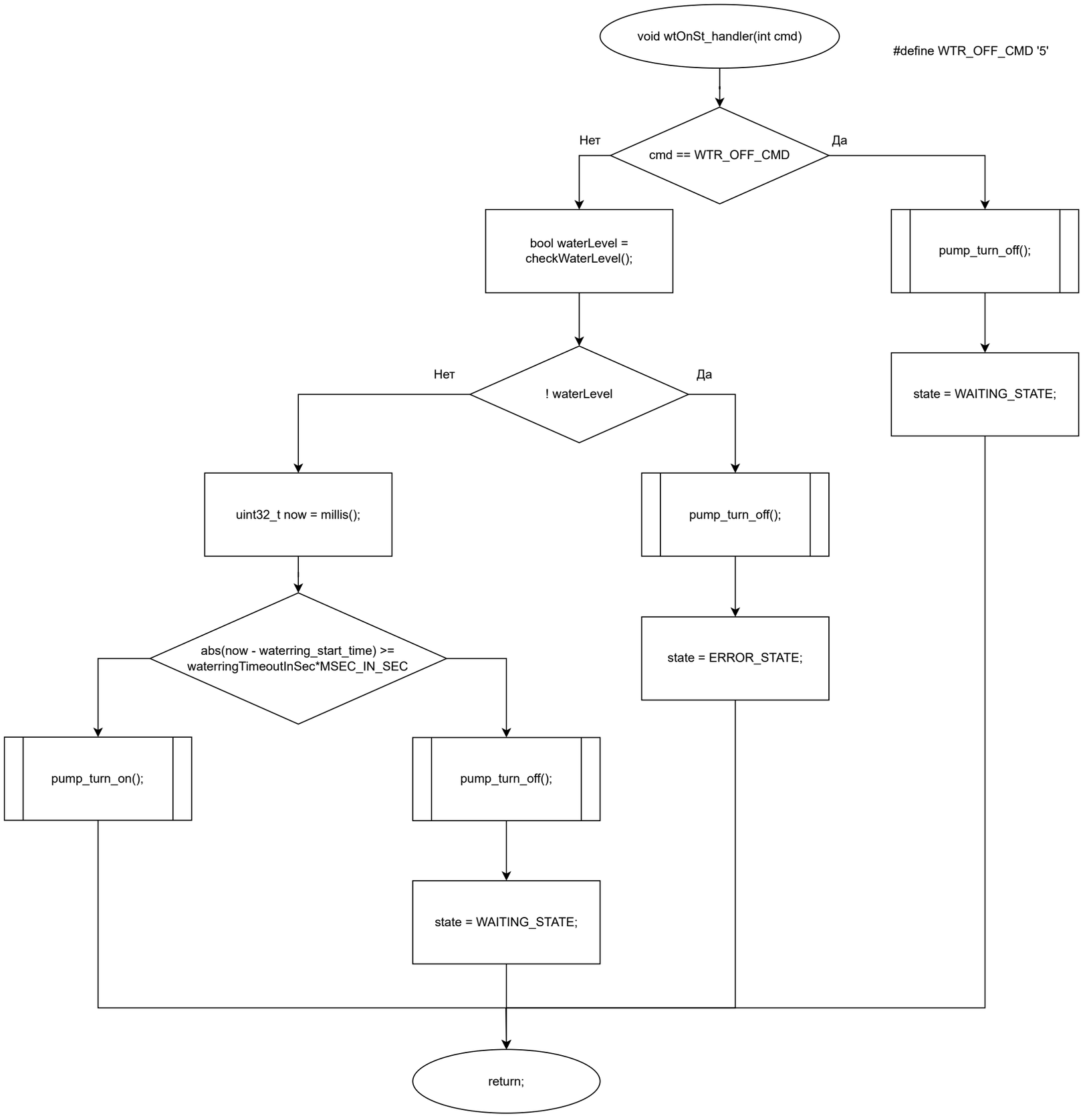
waterring_start_time = millis();

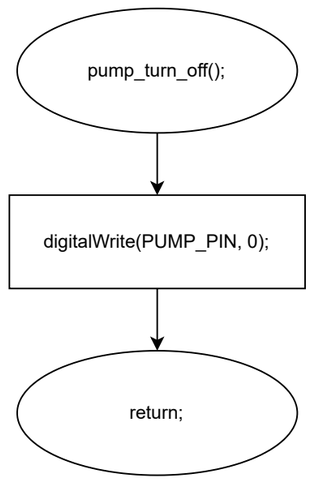
state = WATERING_ON_STATE;

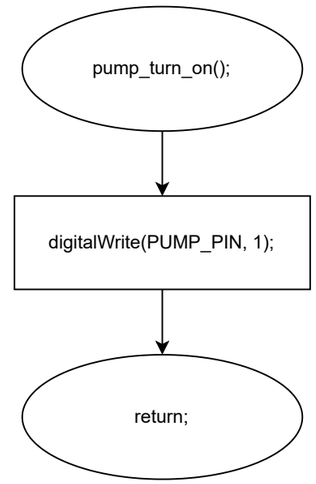
void chWtSt_handler()

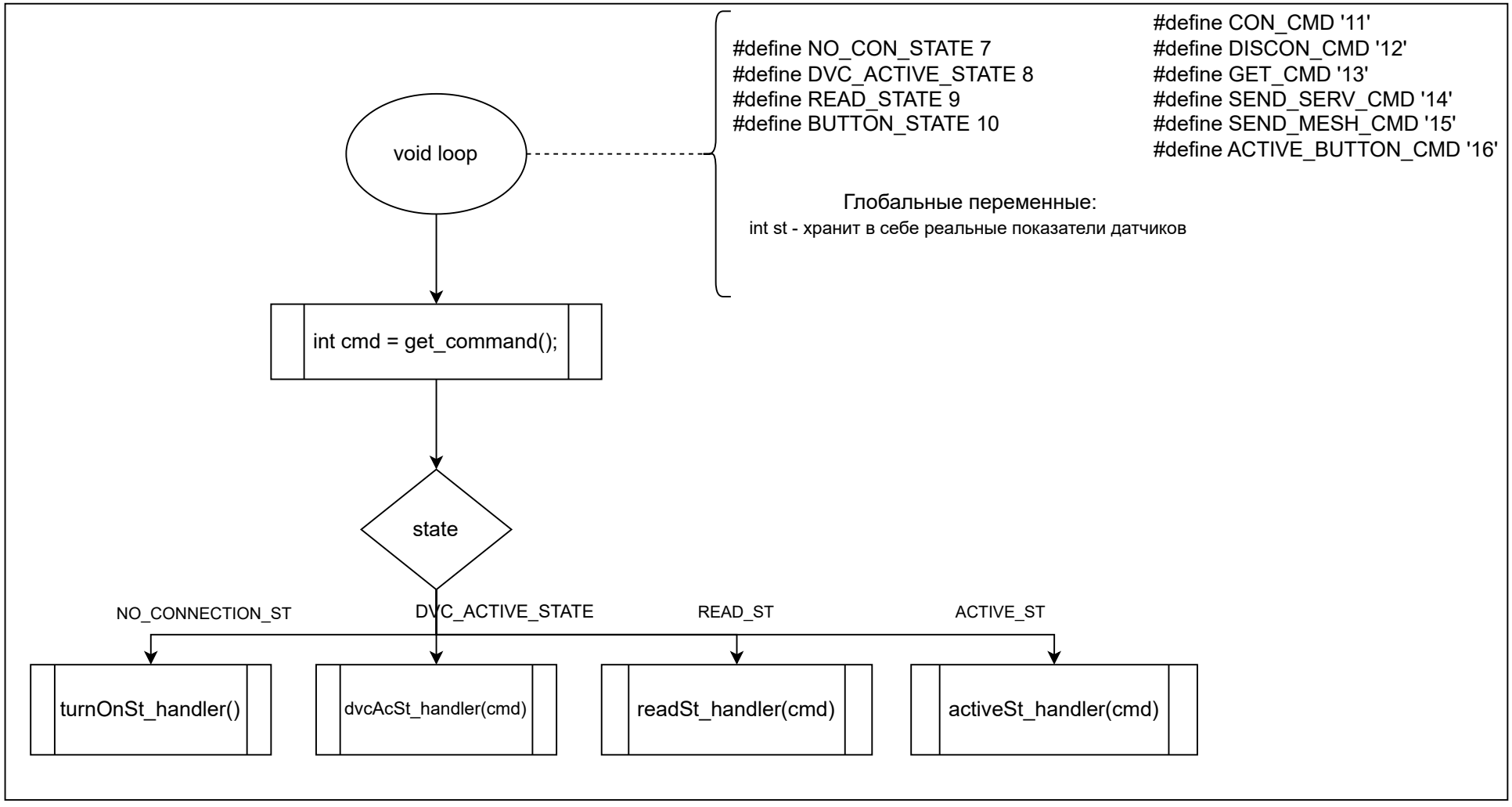
#define MSEC_IN_SEC 1000
int waterringTimeoutInSec = 10;

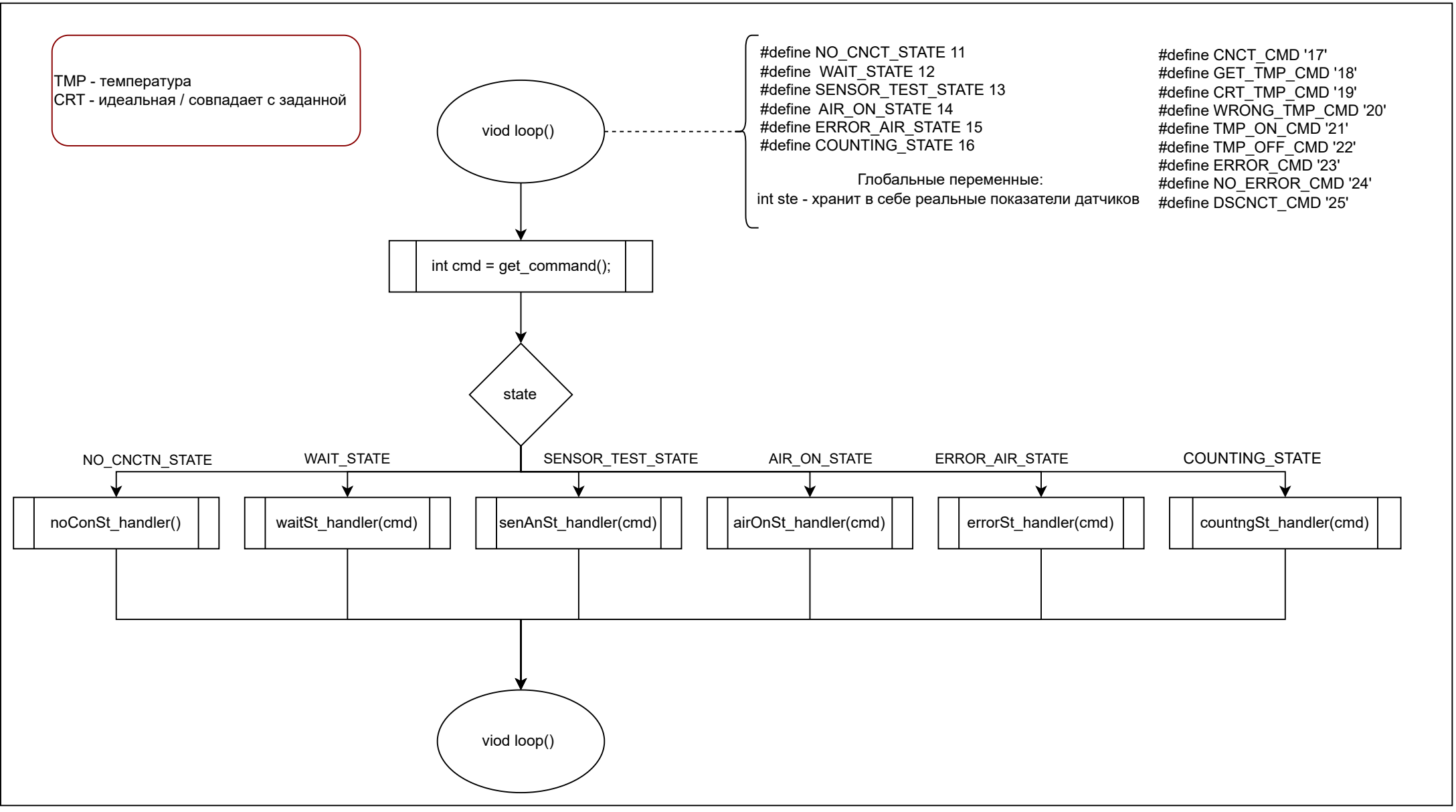
#define WTR_OFF_CMD '5'

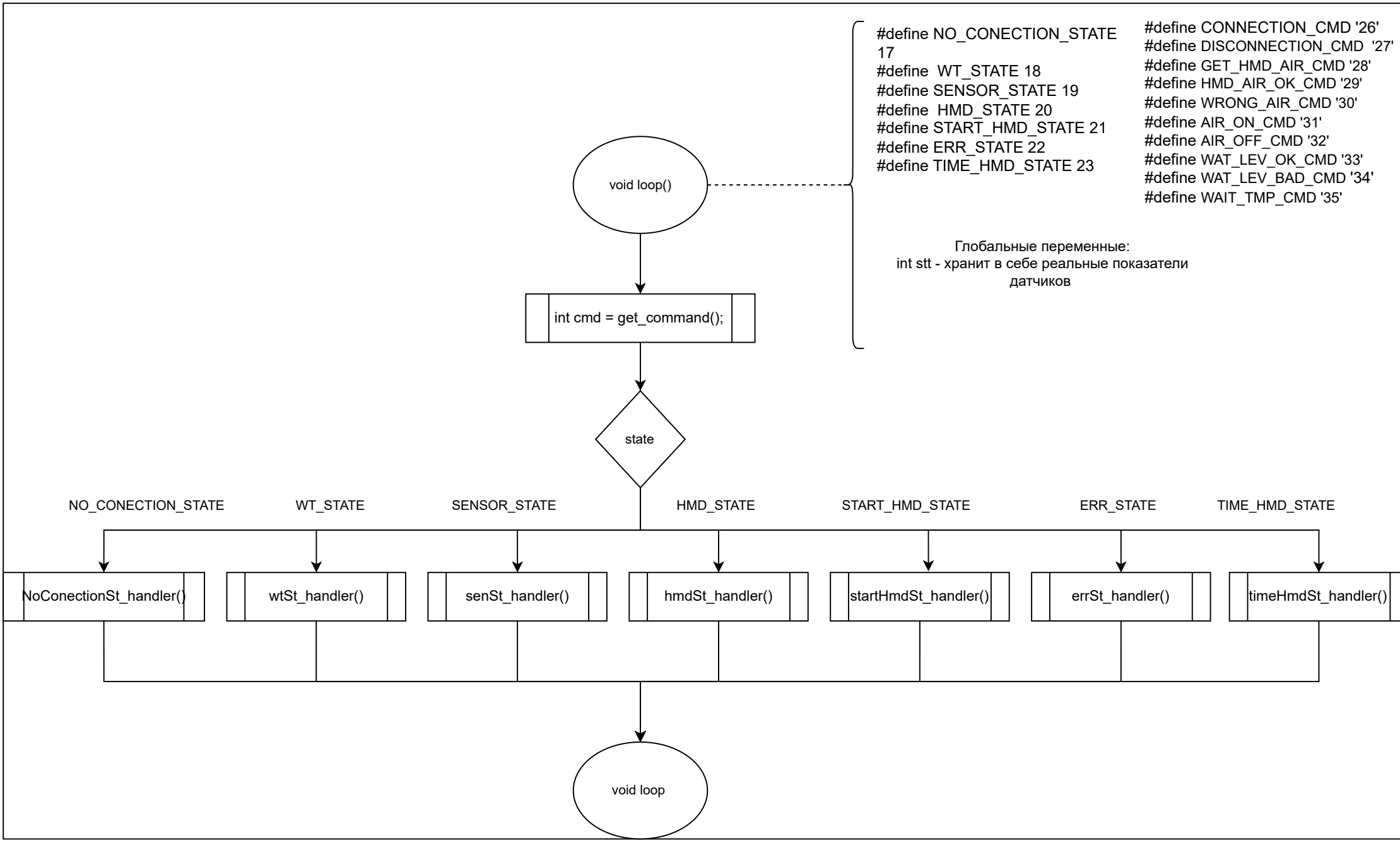







#define NO_CON_STATE 7
#define DVC_ACTIVE_STATE 8
#define READ_STATE 9
#define BUTTON_STATE 10
#define CON_CMD '11'
#define DISCON_CMD '12'
#define GET_CMD '13'
#define SEND_SERV_CMD '14'
#define SEND_MESH_CMD '15'
#define ACTIVE_BUTTON_CMD '16'
Глобальные переменные:
int st - хранит в себе реальные показатели датчиков
void loop
int cmd = get_command();
state
NO_CONNECTION_ST
DVC_ACTIVE_STATE
READ_ST
ACTIVE_ST
turnOnSt_handler()
dvcAcSt_handler(cmd)
readSt_handler(cmd)
activeSt_handler(cmd)


TMP - температура
CRT - идеальная / совпадает с заданной
viod loop()
#define NO_CNCT_STATE 11
#define WAIT_STATE 12
#define SENSOR_TEST_STATE 13
#define AIR_ON_STATE 14
#define ERROR_AIR_STATE 15
#define COUNTING_STATE 16
Глобальные переменные:
int ste - хранит в себе реальные показатели датчиков
#define CNCT_CMD '17'
#define GET_TMP_CMD '18'
#define CRT_TMP_CMD '19'
#define WRONG_TMP_CMD '20'
#define TMP_ON_CMD '21'
#define TMP_OFF_CMD '22'
#define ERROR_CMD '23'
#define NO_ERROR_CMD '24'
#define DSCNCT_CMD '25'
int cmd = get_command();
state
NO_CNCTN_STATE
WAIT_STATE
SENSOR_TEST_STATE
AIR_ON_STATE
ERROR_AIR_STATE
COUNTING_STATE
noConSt_handler()
waitSt_handler(cmd)
senAnSt_handler(cmd)
airOnSt_handler(cmd)
errorSt_handler(cmd)
countngSt_handler(cmd)
viod loop()


#define NO_CONECTION_STATE 17
#define WT_STATE 18
#define SENSOR_STATE 19
#define HMD_STATE 20
#define START_HMD_STATE 21
#define ERR_STATE 22
#define TIME_HMD_STATE 23
#define CONNECTION_CMD '26'
#define DISCONNECTION_CMD '27'
#define GET_HMD_AIR_CMD '28'
#define HMD_AIR_OK_CMD '29'
#define WRONG_AIR_CMD '30'
#define AIR_ON_CMD '31'
#define AIR_OFF_CMD '32'
#define WAT_LEV_OK_CMD '33'
#define WAT_LEV_BAD_CMD '34'
#define WAIT_TMP_CMD '35'
Глобальные переменные:
int stt - хранит в себе реальные показатели датчиков
void loop()
int cmd = get_command();
state
NO_CONECTION_STATE
WT_STATE
SENSOR_STATE
HMD_STATE
START_HMD_STATE
ERR_STATE
TIME_HMD_STATE
NoConectionSt_handler()
wtSt_handler()
senSt_handler()
hmdSt_handler()
startHmdSt_handler()
errSt_handler()
timeHmdSt_handler()
void loop