```
#ifndef GLOBALS_H
#define GLOBALS_H

//TIMING PARAMETERS
unsigned long LOOP_TIME;
unsigned long CURRENT_MILLIS = 0; //used to keep track of time
unsigned long PREVIOUS_MILLIS = 0; //used to keep track of time
unsigned long PREVIOUS_MILLIS_1 = 0; //used in case 8
unsigned long PREVIOUS_MILLIS_2 = 0; //used in case 7
unsigned long PREVIOUS_MILLIS_3 = 0; //used in readTCs function
unsigned long PREVIOUS_MILLIS_4 = 0; //used in readOut function
unsigned long PREVIOUS_MILLIS_5 = 0; //used in blinkGRN() function
unsigned long PREVIOUS_MILLIS_6 = 0; //used in blinkAMB() function
unsigned long PREVIOUS_MILLIS_7 = 0;//used in ignition state to check temps
unsigned long PREVIOUS_MILLIS_8 = 0;//used in
unsigned long LOOP_MILLIS = 0; //used to caluclate loop function time.

//timing constants//valatile because can be changed by dashboard
volatile long SUPERHEAT_TIMER = 5000;
volatile long BMM_OFF_TIMER = 30000;
volatile int BMM_START_TIMER = 5000;
volatile int BMM_PURGE_TIMER = 30000;
volatile long BMM_IGNITION_TIMER = 25000;
volatile long BURNER_RAMP_TIMER = 240000; //4mins
volatile long BURNER_REACH_END_TIMER = 3600000; //SECONDS//1hr
volatile long STEAM_GENERATION_TIMER = 1800000; //30 mins
volatile int STEAM_AT_170PSI_TIMER = 10000; //10 seconds
volatile long OPEN_SR_FUEL_TIMER = 180000; //3 mins

//counter
uint8_t TC_CHECK_COUNTER = 1; //used in readTC function
uint8_t READOUT_COUNTER = 1; //used in readOut function

//BLOWER VARIABLES
volatile int BLOWER_PURGE_SPEED = 5000, BLOWER_IGNITION_SPEED = 800, BLOWER_RAMP_BEGIN = 1500, BLOWER_RAMP_END =
4100; //2 to keep the dps triggered .17 * 20043counts
volatile int BLOWER_TOP_SPEED = 6900;
uint16_t BLOWER_SPEED_AT_170PSI, BLOWER_SPEED_FEEDBACK = 0; //feedback in counts

//RO PUMP VARIABLES
const int MAX_RO_PRESSURE = 250; //psi
volatile int RO_PUMP_AT_10_GRAMS_PER_SEC = 3801 , RO_PUMP_TOP_SPEED = 6000; //counts//3000
uint16_t RO_PUMP_FEEDBACK = 0;
float RO_PUMP_COUNT=0;

//FLOW CONTROL VALVES
uint8_t FCV205_HX406_INLET_FLOW = 0; //g/sec
volatile int FCV205_AT_50_PERCENT = 5000; //initiated in case 9
volatile int FCV205_AT_35_PERCENT = 0; //initiated in case 3??
volatile int FCV205_OFFSET = 0;
uint32_t FCV134_BURNER_FUEL_FLOW_FB;//counts
volatile int FCV134_BURNER_FUEL_FLOW_IGNITION = 5900, FCV134_BURNER_FUEL_FLOW_RAMP_END = 6000; //5 volts ~40 percent
open
volatile int FCV134_BURNER_FUEL_FLOW_RAMP_BEGIN = 6000;
volatile int FCV141_SR_FUEL_START_PERCENT = 2000; //In counts .2*20000
uint32_t FT132_NG_FEED_FLOW = 0; // In grams per second
volatile int FT132_PIPE_DIA_CONV = 0.5;
volatile int FT132_COUNTS_TO_G_PER_SEC = .27545;
volatile int FT132_4MA_OFFSET = -3921;
float FT132_ADJUSTED_MEASURE = 0;


uint8_t ATMOSPHERIC_PRESSURE = 7; //psi

//INITIALIZE PRESSURE TRANSDUCER VARIABLES
uint16_t PT213_RO_PRESSURE, PT318_HX406_OUTPUT_PRESSURE; //psi
uint16_t PT420_STEAM_EJECTOR_PRESSURE, PT304_TWV308_INPUT_PRESSURE; //psi

//INITIALIZE THERMOCOUPLE VARIABLES
uint16_t TT142_SR_FUEL, TT301_HX406_STEAM_OUT, TT303_HX504_STEAM_OUT, TT306_EJECTOR_STEAM_IN;
uint16_t TT313_HX402_STEAM_OUT, TT319_HX402_STEAM_SYSTEM, TT407_STEAM_REFORMER_OUT_LREF;
uint16_t TT408_HTS_IN_LREF, TT410_HTS_OUT_LREF, TT411_FPZ_OUT_LREF, TT430_SMR_TUBES_INLET;
uint16_t TT511_SILICON_CARBIDE_OUT, TT512_SILICON_CARBIDE_OUT, TT513_HX504_IN, TT514_HX504_OUT;
uint16_t TT441_SMR_TUBE1_OUT, TT442_SMR_TUBE2_OUT, TT443_SMR_TUBE3_OUT, TT444_SMR_TUBE4_OUT;
```

```
uint16_t TT445_SMR_TUBE5_OUT, TT446_SMR_TUBE6_OUT, TT447_SMR_TUBE7_OUT, TT448_SMR_TUBE8_OUT;
uint16_t TT449_SMR_TUBE9_OUT;
volatile int BURNER_TEMP_RAMP_END = 850, BURNER_TEMP_CROSSOVER = 880;
uint16_t old_TT511, current_TT511;

//OCI417 MODBUS RS485 PARAMETERS
const int OCI_INPUT_STATUS_REGISTER = 25;
const int OCI_OUTPUT_STATUS_REGISTER = 26;
const int OCI_UNIT_ID = 1;
uint16_t OCI_INPUT_STATUS_WORD = 0;
uint16_t OCI_OUTPUT_STATUS_WORD = 0;
uint8_t OCI_RESULT;
bool COMBUSTION_PRESSURE_SWITCH, BMM_ALARM_STATUS;
bool BMM_PROOF_OF_FLAME, OCI_TO_BMM_COM;
bool DUN_PSH, DUN_PSL, DUN_ZSL;

//FLAGS
bool GRN_BTN_FLAG = false, AMB_BTN_FLAG = false, ESTOP_FLAG = false, SENSOR_INTEGRITY_CHECK = false;
bool GRN_PLT_STATE = false, AMB_PLT_STATE = false;
uint16_t LAST_DI_STATUS_WORD = 0, CURRENT_DI_STATUS_WORD = 0, DI_STATUS_CHANGE = 0;
bool PSI_INIT_TIMER = false;

enum {INITIALIZE, DEPRESSURIZE, SUPERHEAT_TEST, BMM_OFF, BMM_ON,
      BMM_PURGE, BMM_IGNITION, BURNER_RAMP, STEAM_GEN, OPEN_SR_FUEL,
      IDLE_MODE, STABILIZE_MODE
     } FSM_STATE;
//PIDs
int16_t BLOWER_SPEED_OFFSET, RO_SPEED_OFFSET, SR_FUEL_OFFSET;
uint16_t BURNER_FUEL_CUT_OFFSET, BURNER_FUEL_CUT;//for pid in case 9 open sr fuel also case 10
float SR_FUEL_CUT = 0.3; //grams per second

//telemetry
volatile int regRX = 0;

uint8_t ERROR = 0; //ERROR 0 means no ERROR.

String FSM_STATE_STRING;

#endif
```