Embedded System Course 2017-18

C4µC

C/C++ Programming for Microcontrollers

Exercises

Lecturers: Ing. Marco Santic, Ing. Walter Tiberti

marco.santic@univaq.it walter.tiberti@graduate.univaq.it

Exercise 3

- A company that produces engine-generators has asked you to develop a small µC system for the automatic starting:

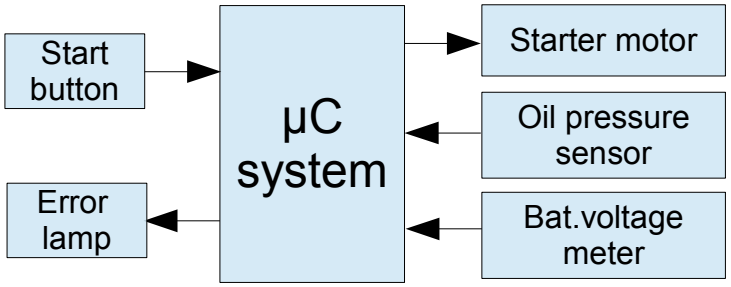
  - The engine-generator is composed by:
    - The electrical generator
    - The engine (the prime mover) with its components (fuel tank, cooling system, ...)
    - A battery
    - An electric starter motor
  - Other components, such as the constant engine speed regulator (governor) and the generator voltage regulator, have already their own mechanical and electrical control systems
  - When the engine is on, the electrical generator charges the battery

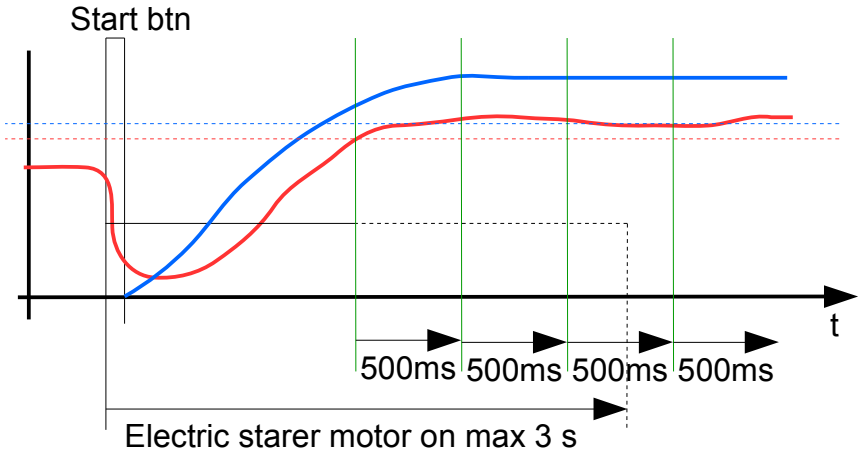
Exercise 3

- Inputs, outputs and requirements:
  - The start is controlled by a START_BUTTON
  - The electrical starter motor must be turned on, immediately after the START_BUTTON is pressed, for a max time of 3 seconds
  - There are two variables to control:
    - The OIL_PRESSURE of the engine (analog signal)
    - The VOLTAGE on the battery (analog signal)
  - If within 3 seconds (starter) the OIL_PRESSURE and the VOLTAGE are above some specified thresholds, the engine is on, otherwise an ERROR should be signaled
  - If, during the engine on-phase, one of the two variables goes below its threshold, the ERROR should be signaled too (check period: 500 milliseconds)

Exercise 3

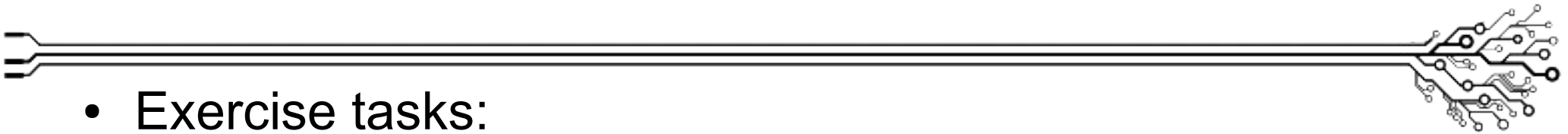
- The system block and time diagram of the signals are:





- Conversion units
  - Oil threshold is 3 bar (analog signal: 0-5V for 0-5 bar)
    - On ADC, with a precision of 8bit and 5V full scale, is "154"
  - Voltage threshold is 13V (analog signal: 0-5V for 5-15V)
    - On ADC, with a precision of 8bit and 5V full scale, is "205"

Exercise 3



- Exercise tasks:
  - Define inputs and outputs for the reference μC 328p
  - Draw the state-chart or the control-flow diagrams for the system
  - Write down pseudocode for the init function (e.g. s*etup()*) and the main function (e.g. *loop()*), declaring all the needed variables and keeping separated the functions for the reading of analog inputs
  - Write any comment to justify a choice when the system requirements leave a degree of uncertainty

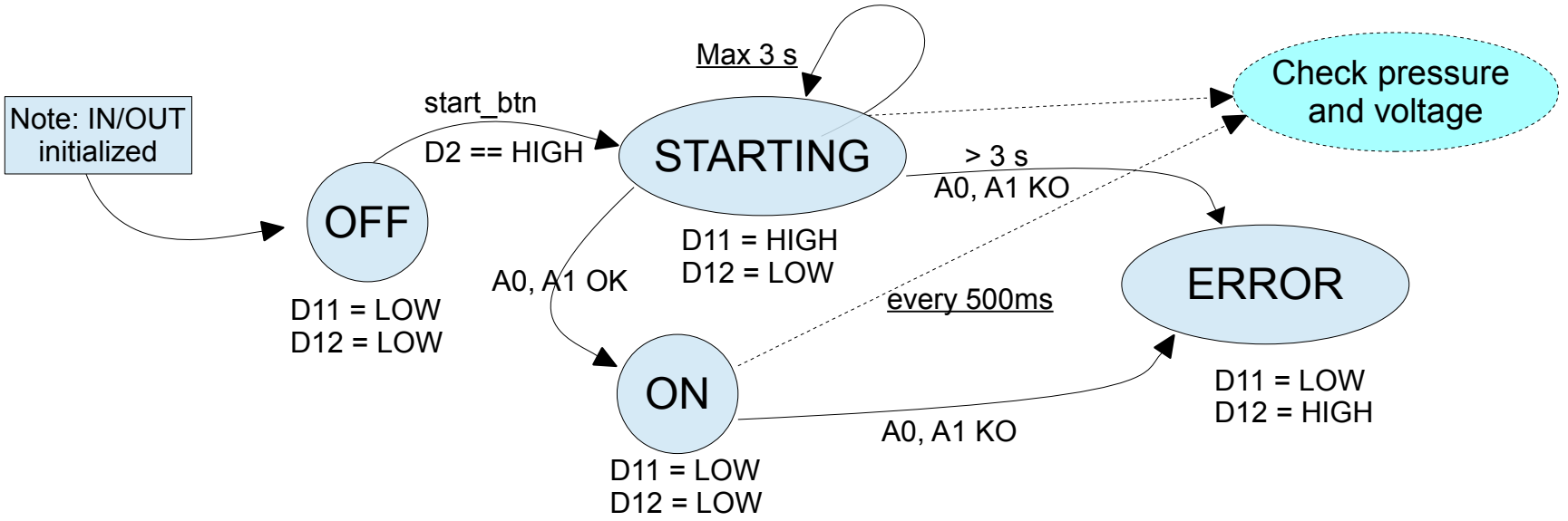
Solution 3

- The inputs/outputs of the system can be:

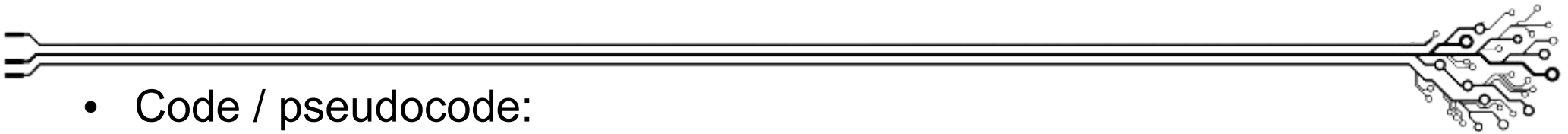
Signal	IN / OUT	Pin assignment	Notes
START	Digital IN	D2	HIGH when pressed
ST_MOTOR	Digital OUT	D11	HIGH for ON
OIL_PRESSURE	Analog IN	A0	>= 154
V_BATTERY	Analog IN	A1	>= 205
ERROR	Digital OUT	D12	HIGH for ON

Note: not defined in requirements

- The state chart can be:



Solution 3



- Code / pseudocode:

```
// definitions
#define BTN_START      2
#define EL_STARTER_M  11
#define OUT_ERROR     12
#define OIL_P         A0
#define BAT_V         A1

#define ST_OFF         0
#define ST_STARTING    1
#define ST_ON          2
#define ST_ERROR       3



// global vars
uint8_t sys_state;

// init inputs, outputs, ...
void init_hw_sw(){
  pinMode(BTN_START, INPUT);
  pinMode(EL_STARTER_M, OUTPUT);
  pinMode(OUT_ERROR, OUTPUT);
  digitalWrite(EL_STARTER_M, LOW);
  digitalWrite(OUT_ERROR, LOW);
  sys_state = ST_OFF;
}
```

Solution 3

- Code / pseudocode:

```
// check_function
uint8_t check_oil_and_voltage(){
  uint8_t pressure_value = analogRead(OIL_P);
  uint8_t voltage_value = analogRead(BAT_V);
  if((pressure_value >= 154) & (voltage_value >= 205)) return 1;
  return 0;
}
// main function
int main(){
  init_hw_sw();
  // after init, we are already in OFF state, waiting for START button pressed
  while(!digitalRead(BTN_START));    // blocking until pressed
  sys_state = ST_STARTING;           // so, let's set the outputs...
  digitalWrite(EL_STARTER_M, HIGH);
  unsigned long int time_stop = millis() + 3000;
  while(millis() < time_stop){
    if(check_oil_and_voltage()){
      sys_state = ST_ON;             // pressure and voltage OK
      break;                         // exit the while
    }
  }
  digitalWrite(EL_STARTER_M, LOW);   // state ST_ON or ST_ERROR, both with starter off
  if(sys_state == ST_ON){
    do {                             // should never exit from this do-while
      delay(500);
    } while(check_oil_and_voltage());
  }
  sys_state = ST_ERROR;
  digitalWrite(OUT_ERROR, HIGH);
  return 0;
}
```