PART N° 3 Software description

10071 Borgaro T.se (TO) - Via Mappano, 17 - T +39 011 451 8638 r.a. - F +39 011 470 4891

The Arduino IDE (integrated development environment) contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions, and a series of menus. It connects to the Arduino hardware to download programs and communicate with them.

At the following official link is possible a free Download of the Arduino Enviroment Software

http://arduino.cc/en/Main/Software

10071 Borgaro T.se (TO) - Via Mappano, 17 - T +39 011 451 8638 r.a. - F +39 011 470 4891

Some considerations must be done before start to write the program.

- A sw program is a list of instructions executed sequentially.
- More instruction are, more time is requested to do a complete loop, to return at start point.
- When you want control a physic event like the movement of the actuator, more short is the time of the sw loop, more precise is the controlling and where it is.

A compromise may be found if you organize the sw into high and low speed activities.
In this manner is possible executing at high frequency the functions that requires real time control and in the rest of the time, subdivided in more steps may be executed the non critical functions.

In this project the software has been organized in tasks controlled by a simple operating system synchronized with interrupt TIMER1.
With this method has been possible to assign to each task ,different priority and execution time, following the importance of the functions

10071 Borgaro T.se (TO) - Via Mappano, 17 - T +39 011 451 8638 r.a. - F +39 011 470 4891

In this program the SW manages :

- Some routines interrupt controlled,
- One main loop (free run)
- Eleven tasks real time, controlled by Timer1.

One of these the TASK0, is executed at every Timer1 interrupt. It contains the time critical functions
For the others at each interrupt, a counter is incremented and its value indicates the task that must be processed in that time step. The operations associated are executed one time every ten steps.

The instructions executed at every interrupt routine (pulses decoding and the tasks of counting) require about 25 micro seconds.
The remain time (500 - 25 = 475 us) is available to execute main loop, TASK0 and one of the other tasks.

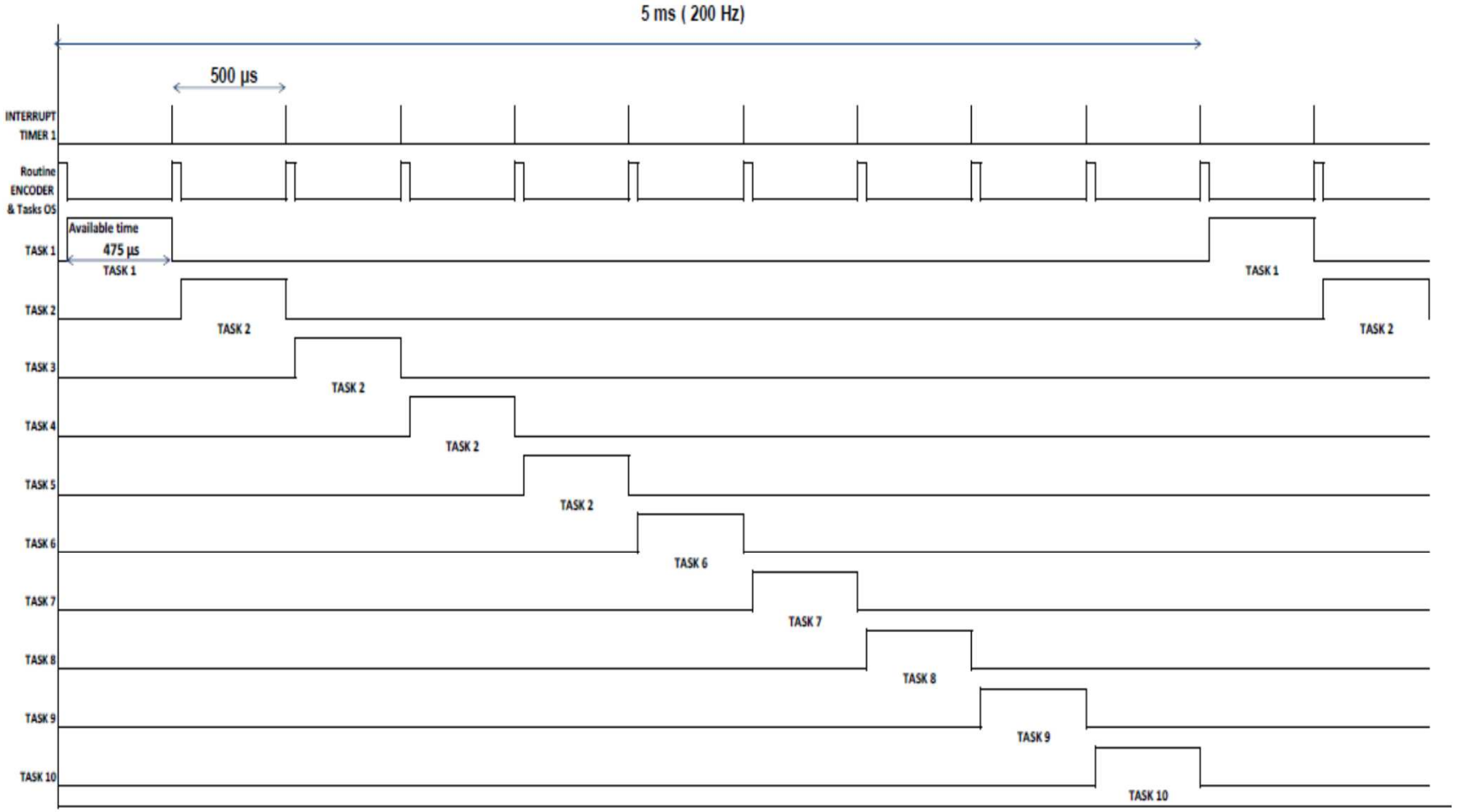
The assumption is that Task 0 uses only a part of this time and the rest of the time, is sufficient to execute the instructions of the active TASK (1 to 10) in that step.

With this method TASK0 is executed at 2 kHz, like for the pulses counting. The other TASKs (1 to 10) at 200 Hz.

The measured typical time, from the interrupt TIMER1 to end of task Manager without active functions is of 40/50 us
The same interval with the (actual) basic control function active is about 100 us.
This means that for new functions will be available about 350/400 us.

10071 Borgaro T.se (TO) - Via Mappano, 17 - T +39 011 451 8638 r.a. - F +39 011 470 4891

Software timing



10071 Borgaro T.se (TO) - Via Mappano, 17 - T +39 011 451 8638 r.a. - F +39 011 470 4891

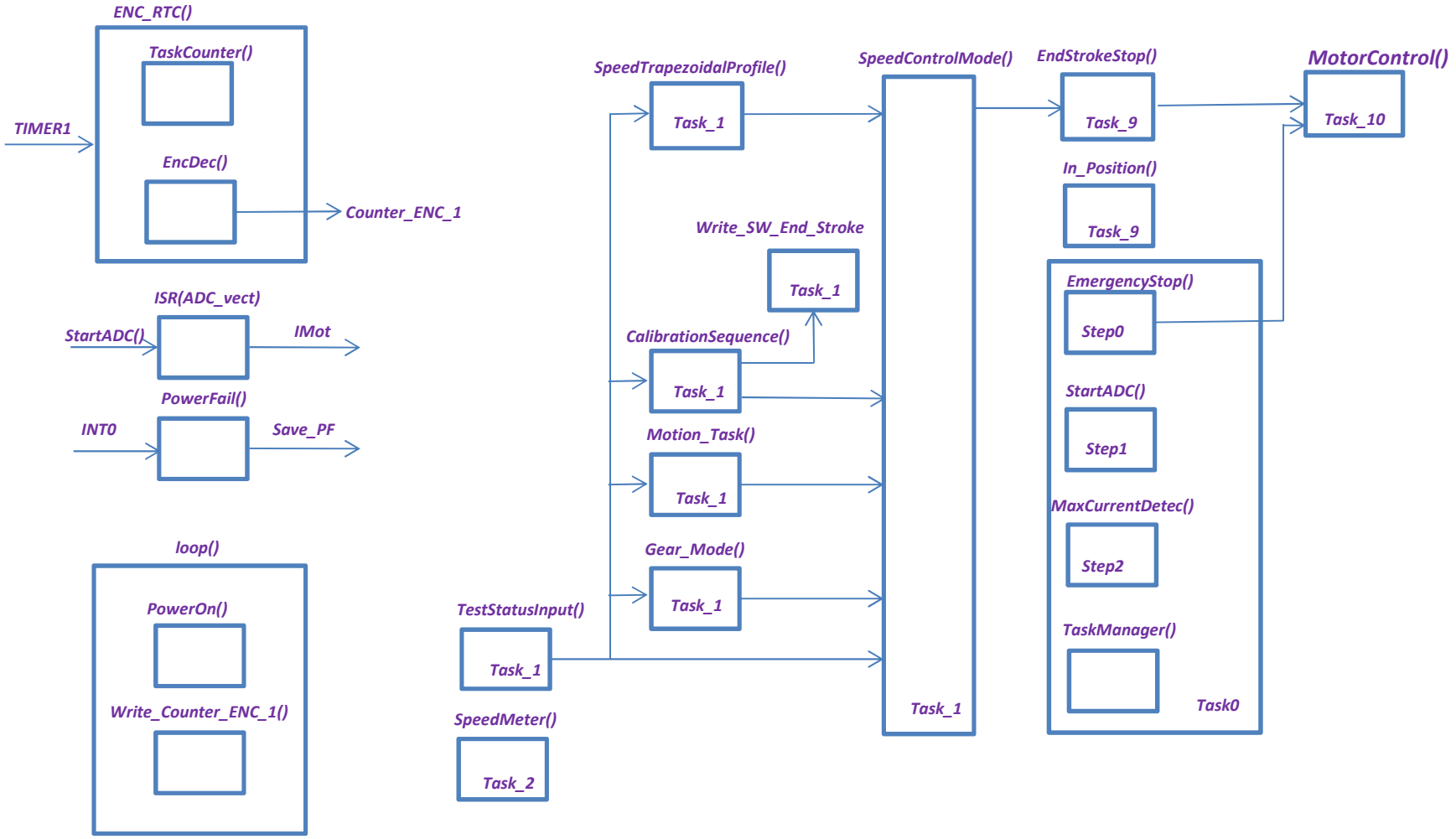
To a readable document, in the sw project each activity has been written as an independent "Function“.
This permits a better and more easy reading of the program instructions and monitoring.
Every function may be documented also as a flow chart and as a graphical function block .
Is possible obtain a simple representation of the entire program, connecting these functions block in a schematic block drawing.

On the www.setec-group.it are available click_here:

- The flow charts of the single functions.
- The last version of the prototype program (august 2013) : Servosystem_LACx4_V_4_Motion.ino

10071 Borgaro T.se (TO) - Via Mappano, 17 - T +39 011 451 8638 r.a. - F +39 011 470 4891

Motor Control functions block



10071 Borgaro T.se (TO) - Via Mappano, 17 - T +39 011 451 8638 r.a. - F +39 011 470 4891

List of the library and the main routine of the program

The following libraries must be included in the program:

#include "***TimerOne.h***"	This library must be downloaded from Arduino site. It permits to modify the time base of the real time clock TIMER1
#include <***EEPROM.h***>	This library is present in the standard IDE. Using the instruction EEPROM.write(Address, byte); is possible to write and save one byte at the specified address Be careful the operation takes 3,3 ms (each byte) to be completed

The routines of this prototype are:

Interrupt activated

- ***ENC_RTC()***. This is launched by interrupt TIMER1. It executes, ***EncDec()*** that captures and counts X4 the encoder transitions and ***TaskCounter()*** that controls the step of the task manager.
- ***PowerFail()***. This is launched from the high to low transition of the PowerFail signal connected to the INT0 (***PIN 8 D3/SCL PF_INT)*** It enables "one shot" execution of ***Write_EEPROM()*** function during power down.
- ***ISR(ADC_vect)*** It is activated automatically at every end conversion of ADC. To be executed requires a start conversion command.

Free run functions

- ***loop()*** The included instructions are executed in free run mode. The loop is in paused only during the interrupt events. The loop manages also the launch of the Tasks enabled.

Functions event controlled

- ***PowerOn()*** Executed "one shot" at power on
- ***Write_Counter_ENC_1()*** During the power down, this function save actual actuator position.

The sum of the execution time of the simultaneous interrupt + free run functions + event controlled functions + task functions, must be less of 500 micro seconds, to avoid a processing overrun problems.

10071 Borgaro T.se (TO) - Via Mappano, 17 - T +39 011 451 8638 r.a. - F +39 011 470 4891

The Task manager:

- ***TaskManager()*** This task (TASK0) is enabled under interrupt ENC_RTC). It runs “one shot” every 500 micro seconds. In this task are made two steps. One executes sub steps of the TASK0. The second jump to the task enabled (from TASK1 to TASK10)
- ***TASK1 to TASK10*** These tasks are enabled sequentially by task manager (under interrupt ENC_RTC).

Functions:

- ***CalibrationSequence ()*** Permits to set and save the limit stroke positions.
- ***EmergencyStop()*** STOP motor. Stop is executed when both INPUT_FWD && INPUT_BKW or when current motor >= Imax.
- ***EndStrokeStop()*** STOP motor with ramps at reaching th sw limits.
- ***In_Position()*** When actuator is inside the position window, signals activating a dedicated output.
- ***MaxCurrentDetec()*** Control if motor current is less of Imax motor limit.
- ***Motion_Task()*** Here are programmed in sequential the motion tasks executed in motion control mode.
- ***MotorControl()*** Controls the output motor power circuits (relays and POWER MOSFET).
- ***SpeedControlMode()*** Controls the source of the speed commands (Operator ON/OFF, Calibration, Motion task, Gear mode).
- ***SpeedMeter()*** Measure actual speed motor revolution. Due to the few hall sensor pulses, for a good resolution the value is supplied every second.
- ***SpeedTrapezoidalProfile()*** Generates a speed trapezoidal profile command reference.
- ***StartADC()*** START the analog to digital conversion.
- ***TaskCounter()*** Controls the task enabled at each timer1 (loop).
- ***TestStatusInput()*** Acquires status of digital inputs and set status of the operator commands.
- ***Write_Counter_ENC_1()*** During the power down, this function save actual actuator position.
- ***Write_SW_End_Stroke()*** At the end of the Calibration function this routine save Positive SW Limit in EEPROM

10071 Borgaro T.se (TO) - Via Mappano, 17 - T +39 011 451 8638 r.a. - F +39 011 470 4891