Arduino controlled sample positioning stage and automatic data acquisition using OMDAQ3

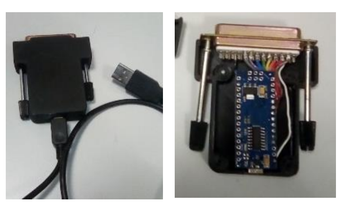
M. Fortunato[1], N. Catarino[2], L.C. Alves[1,3], V. Corregidor[1,3], R.C. da Silva[2,3]

1) C2TN, Instituto Superior Técnico (IST), Campus Tecnológico e Nuclear, Estrada Nacional 10, P-2695-066 Bobadela LRS, Portugal
2) IPFN, Instituto Superior Técnico (IST), Campus Tecnológico e Nuclear, Estrada Nacional 10, P-2695-066 Bobadela LRS, Portugal
3) DECN, Instituto Superior Técnico (IST), Campus Tecnológico e Nuclear, Estrada Nacional 10, P-2695-066 Bobadela LRS, Portugal

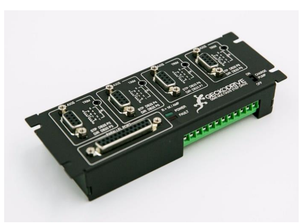
Introduction

In this work a simple, effective and low-cost system is described for actuating a stepper-motor driven x-y stage based on an Arduino Nano board together with a multi axis digital stepper driver module (Geckodrive G540) with commanded positioning and automatic data acquisition under OMDAQ3 environment. .

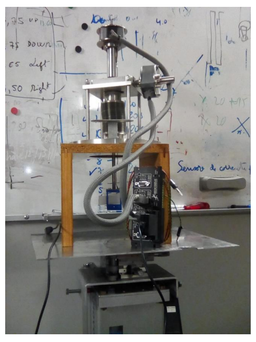
Hardware



Arduino Nano board
USB connection port:
- any type of computer; no need for internal bus cards
DB25 connector:
- Connection to the stepper-motor power driver
3D printed box:
- accommodates both the Arduino board and the DB25 male connector (robustness and ease of connection)



Stepper-motor driver
Geckodrive G540
- Controls up to 4 stepper motors
- DB9 output connector to stepper motors
- DB25 input/output connector
- Power from external PSU (18– 50 V; 3.5 A máx.)
- Short circuit, overvoltage and overcurrent protection

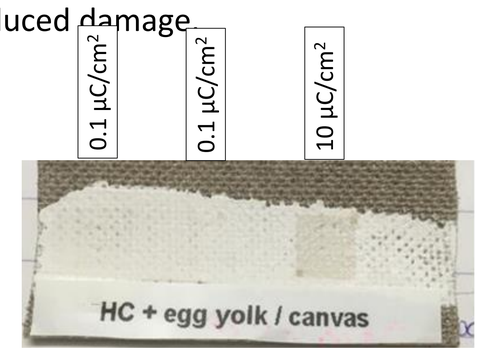


Motorized x,y stage
- 50 mm dislocation amplitude both in x and y
- 4 phase stepper motors RS 318-711 (wired as 2 phase, rating: 12 V, 0.24 A/phase w/ 240 Ω limiting resistor across pins 1 and 5 of DB9 connector)

Application

see more

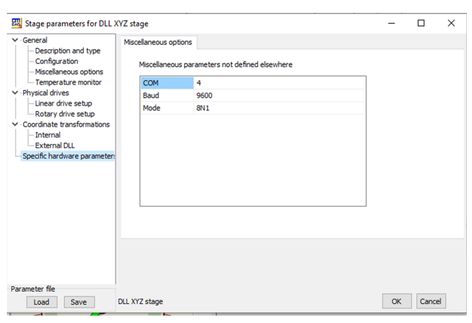
The hardware/software combined system previously shown was used in our external beam setup (1×1 mm^2 exit nozzle window dimensions) for the irradiation of easel painting coupons containing lead white pigments with different binders and supports. The experiment and further analysis required sample irradiated areas of 8×8 mm^2 with accurate measurement of accumulated beam charge per unit area in order to determine exposure thresholds for preventing beam induced damage.

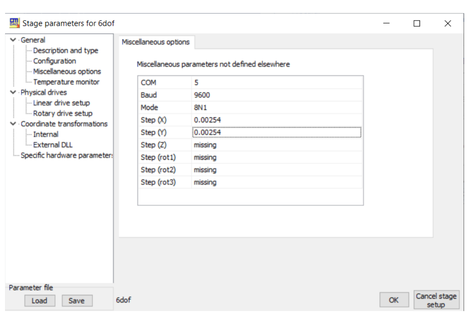


- Ion beam irradiation alteration test on a lead white pigment (lead carbonate form) with an egg yolk binder and painted on canvas.
- 2 MeV proton beam – tested fluences: 0.1 µC/cm^2, 1 µC/cm^2 and 10 µC/cm^2,
- Irradiated area 8x8 mm^2.

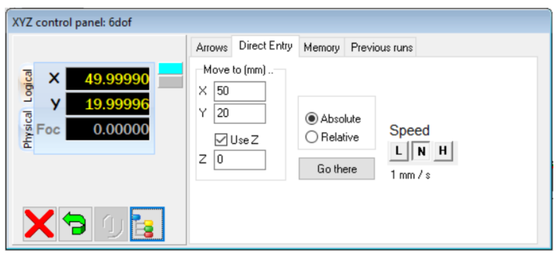
Software

OMDAQ3 interface
- DAQ3 built-in capabilities allow modifying DLL source code for adapting to users motor control interface (allows controlling a system with up to 6 degrees of freedom – 3 translations and 3 rotations).
- Offers USB communication support from PC to Arduino I/O port.
- DAQ3 “hardware options” window allows providing communication parameters (I/O port, baud rate and handshake) together with the definition of motion parameters (e.g. minimum displacement).





- Programmed runs using an EXCEL CSV script file for defining stage to move, scan type (e.g. full scan, point) and data acquisition presets (e.g. time, spectrum count).
- C++ code compiled using Embarcadero’s C++ Builder 10.3.3 Community Edition for obtaining a DLL file (must be named OmXyzDll.dll to be recognized by OMDAQ-3).



Arduino Board Initialization and operation
- Use of modified Arduino open-source libraries
- Waits receiving a string with motion parameters:
“Xd Xs Yd Ys”
That considers 2 linear stages each of them with movement direction (*d)* and number of steps (*s)*.
- Sends to G540, motor direction bit and (s) pulses (digital outputs).
- Gets ‘flag’ from G540 signalling end-of-segment (motion)

Summary

- A low cost, off-the-shelf solution for motorized x,y stage control is devised and provided.
- Rewritten DLL and integration with OMDAQ3 environment using “universal” USB communications with Arduino Nano board.
- System tested for accurate exposure of painting test coupons.

Acknowledgements
The authors, and in particular Manuel Fortunato, would like to thank to Geoff Grime for is support and guidance on the development of the DLL source code

Funding
UIDB/04349/2020
CRP F11021