

Remote control and monitoring system for the FAUST-QTS beam line

M. B. Chapman, P. Cammarata, S. Behling, and S. J. Yennello

Using Arduinos to Remotely Control and Monitor Stepper Motors and Solenoid Actuators

The construction of the new FAUST-QTS beam line has been completed with its inaugural experiment already concluded. One unique aspect of the new beam line is the inclusion of a stepper motor and solenoid actuator control system. The system allows for remote access to control systems inside the cave such as the target ladder position, Faraday cup alignment, and PPAC selection as shown in Fig. 1. Preset positions for the stepper motors can be set and recalled without the need for realignment. This saves wasted beam time by facilitating adjustments to the targets, detectors, and collimators without needing to enter the cave. The system consists of a computer with a USB hub branching off into the different Arduinos [1]. Arduinos are small open source micro controllers chosen for this application because of their low cost and ease of use. These Arduinos act as control and monitoring nodes for either a stepper motor or a solenoid actuator. They report the motors' and actuators' positions back to the computer, while also providing the power to drive them. A Python [2] script displays the information and allows the user to change settings through an intuitive graphical user interface (GUI).

There are two different types of nodes: stepper controller and solenoid controller. Each stepper



FIG. 1. Two stepper motors and two solenoids on top of the FAUST chamber in cave 2B.

controller consists of an Arduino with a motor shield attached that allows the Arduino to move the stepper motor with power provided by a DC power supply. Each solenoid controller is made up of an Arduino that provides a 5V DC signal to a relay board to toggle 120V AC to the solenoid (see Fig. 2). A 5V filter board was needed to prevent current feedback, generated when the solenoid is toggled, from causing the Arduino to lock up. A software reset was also implemented that can restart the Arduino if it does freeze. The GUI was developed utilizing Python with the Tkinter [3] graphics library, and PySerial [4] communications library. The interface allows for remote monitoring and motor movement with savable pre-sets for stepper motor positions. The system has been in use for over a month of beam time.



FIG. 2. A close up of the Arduino inside of the control box.

[1] Arduino, Arduino Micro Controller (<https://arduino.cc>)

[2] M. Lutz and D. Ascher, O'Reilly and Associates (1999)

[3] Tkinter, <https://wiki.python.org/moin/TkInter> (<https://wiki.python.org>)

[4] PySerial, <http://pyserial.sourceforge.net/> (<http://pyserial.sourceforge.net/>)