

DORDT UNIVERSITY

Engineering Department

EGR 304 LAB #3

PARALLEL I/O AND MORE WITH ARDUINO

SPRING 2020

OBJECTIVES

- To use some lower-level I/O instructions to provide byte-wide access to parallel I/O operations.
- To provide analog-like outputs from the Uno to devices that respond well to pulse-width-modulation (PWL).
- To understand analog input capabilities of the Uno.

REFERENCES

- Arduino Reference, Port Registers: <https://www.arduino.cc/en/Reference/PortManipulation>
- Gerald Recktenwald, “Basic Pulse Width Modulation”
http://web.cecs.pdx.edu/~gerry/class/EAS199A/topics/pdf/PWM_output_Arduino.pdf
- Instructable: “How to use a potentiometer” <https://www.instructables.com/id/How-to-use-Potentiometer-Arduino-Tutorial/>

DELIVERABLES

After you get your parallel I/O version of “Cylon Eyes” working please invite the instructor to your workbench and demonstrate your version. Then save your code (in the IDE) as a *.ino file and upload it to Canvas using the link provided in Canvas for this lab.

ACTIVITIES

THE GOAL OF THIS LAB

Today’s main goal is to design a Cylon Eyes display using your Arduino Uno board and using lower-level parallel I/O methods. Use eight LEDs. Your version of Cylon Eyes must have a way to change the speed, the same as in the previous lab. You may use normal `pinMode` and `digitalRead` instructions for input operations.

Please use “Port D” for the digital outputs. This corresponds to digital outputs 0 through 7. You will have to disconnect your LED circuit from pin 0 when you download code into the Uno board. This is because the IDE uses pin 0 as an input. If you have an LED connected to it pin zero will not function properly as an input. For all practical purposes, you may do all your testing and debugging with pin 0 disconnected from its LED and then, when all appears to be working, connect that final LED and test that.

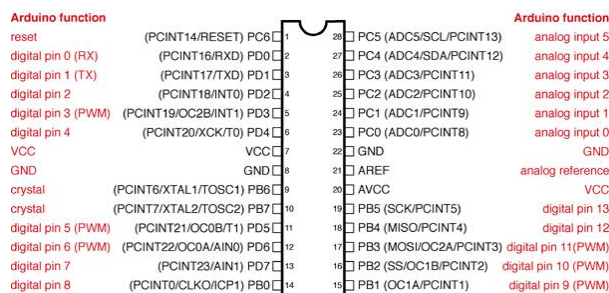


Figure 1, pin mapping on the Arduino UNO board.

The processor’s datasheet uses different labels than the IDE presents to the Arduino programmer. But the processor’s ports and functionality are there in hardware.

After you get Cylon Eyes working with parallel I/O, refer to the handout by Rectenwald. Use the switching hardware from Cylon Eyes to control the brightness of one LED to four levels (full on, medium brightness, low brightness, full off).

Once you get that working, get a DC motor to operate at four speeds. Be sure to use a drive transistor. Type 2N3904 is suitable as is type 2N2222. Also be sure to use a flyback diode. Type 1N4004 is recommended.

Finally, if there is time, experiment with using an analog input pin with a potentiometer in place of the switches to control the motor speed.