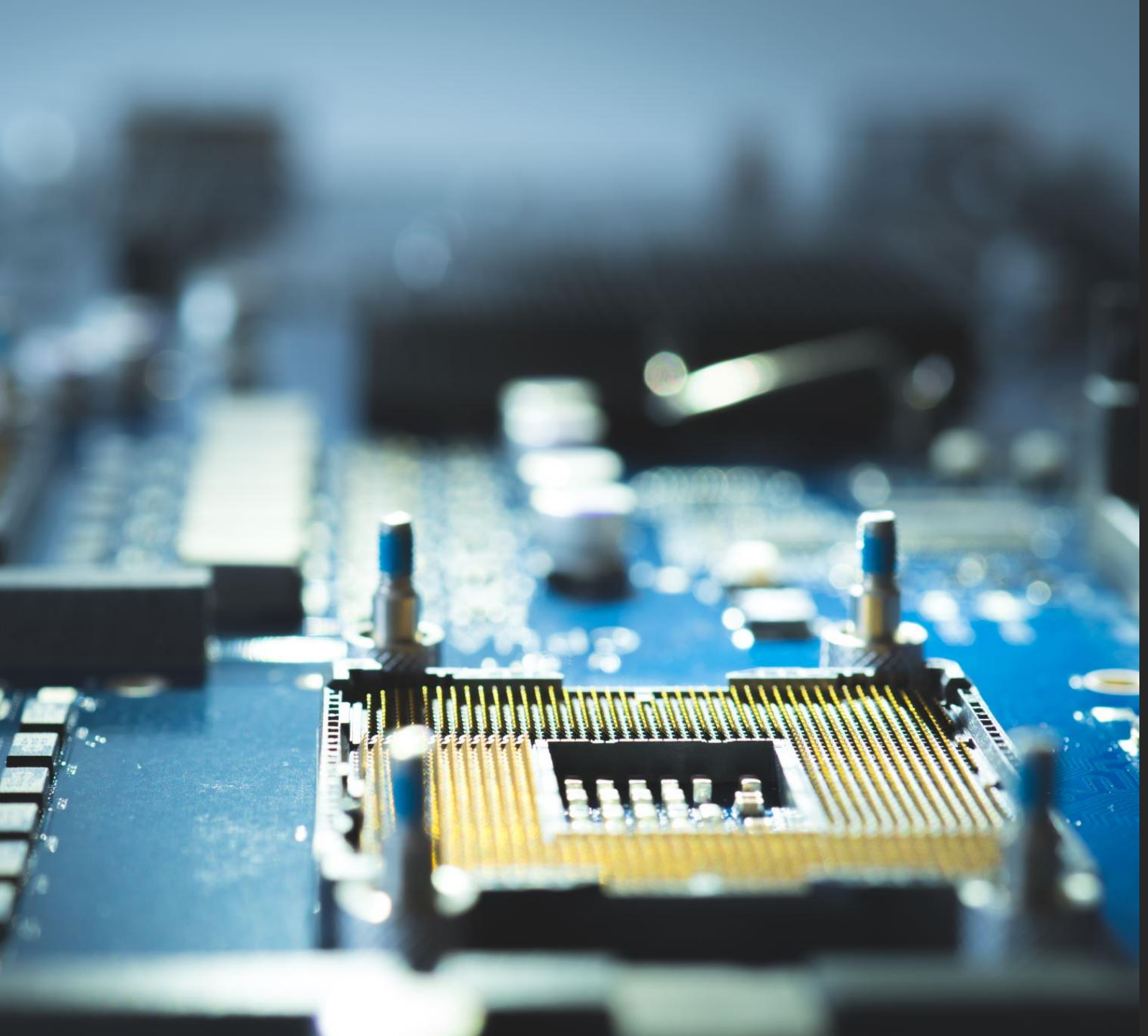


MICROCONTROLLER PWM SIGNAL GENERATOR USER GUIDE

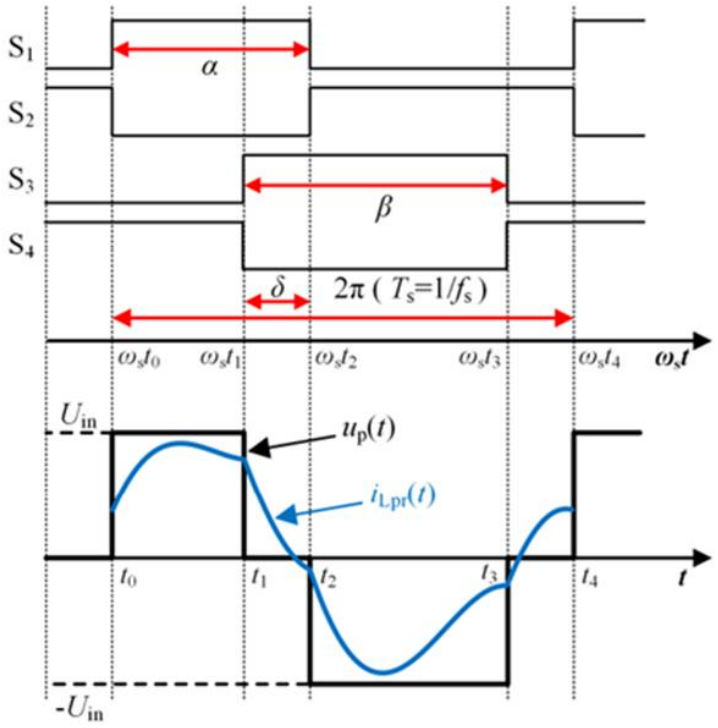
BY: CRAIG SANCHEZ

6/12/2022

OVERVIEW

- Step-by-Step Instructions
  - Background Information
    - Design Objectives
    - PWM Registers
  - Connect Arduino Due USB port of your computer
  - Connect PIN out
  - Open Arduino IDE (Arduino_DUE_PWM.ino)
  - Upload the sketch to Arduino Due (Sketch -> Upload)
  - Use Serial-terminal to change frequency, duty cycle, phase & dead time.
  - Additional PIN-OUT information

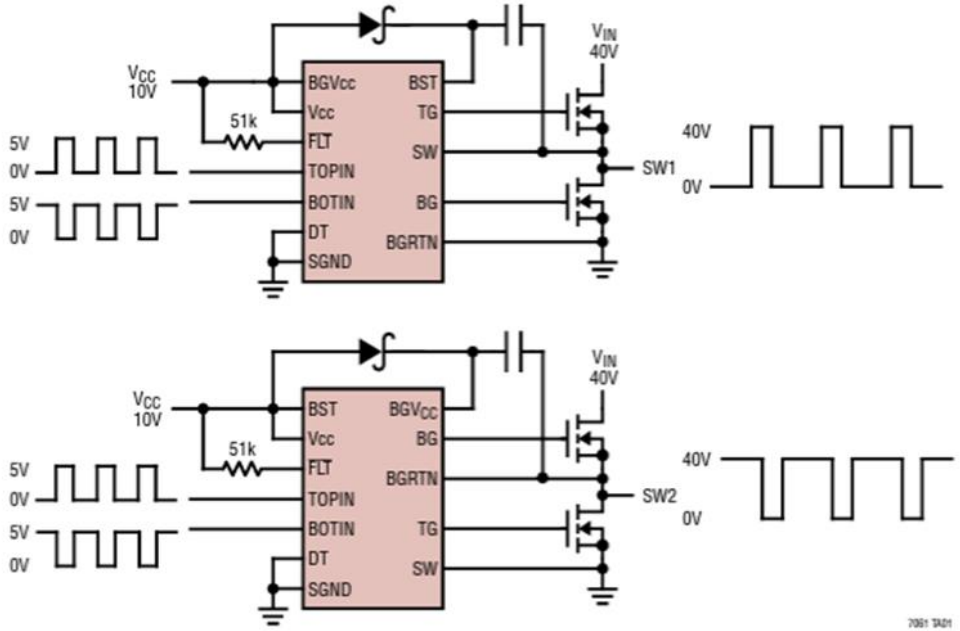
OVERALL DESIGN OBJECTIVES



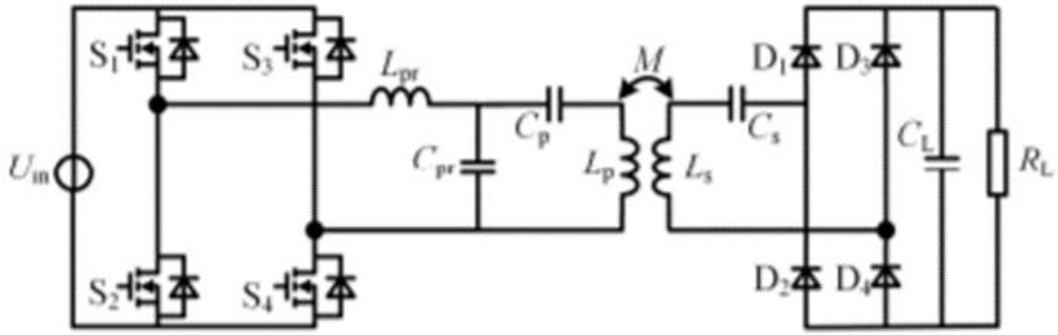
[1] See reference slide for more information.

- The objective of this project is to generate the following signals shown on the figure to left. The signals consist of 4 PWM channels where S1 & S2 are complementary signals. The same can be said for S3 & S4. This means that when one signal is logic HIGH the other is logic LOW. The complementary signals both have the same frequency and duty cycle. The goal is to give the user the ability to change these parameters and control the output signals of the four PWM signals.
- From the timing diagram notice that the duty cycle of complementary pair S1/S2 is inverse to that of S3, S4 therefore when adjusting the duty cycle the result will follow the same logic.
- The main controls that the microcontroller will give to the user via a UI (GUI) is to change the frequency, duty cycle, phase and dead time for all four signals. With the option of adding more PWM channels if desired.

PWM SIGNALS TO GATE DRIVER PROJECT



- The purpose of creating 4 PWM signals is to then feed the signals into a full bridge inverter to generate and AC signals that can then be transmitted wirelessly via inductive coils for power transmission.

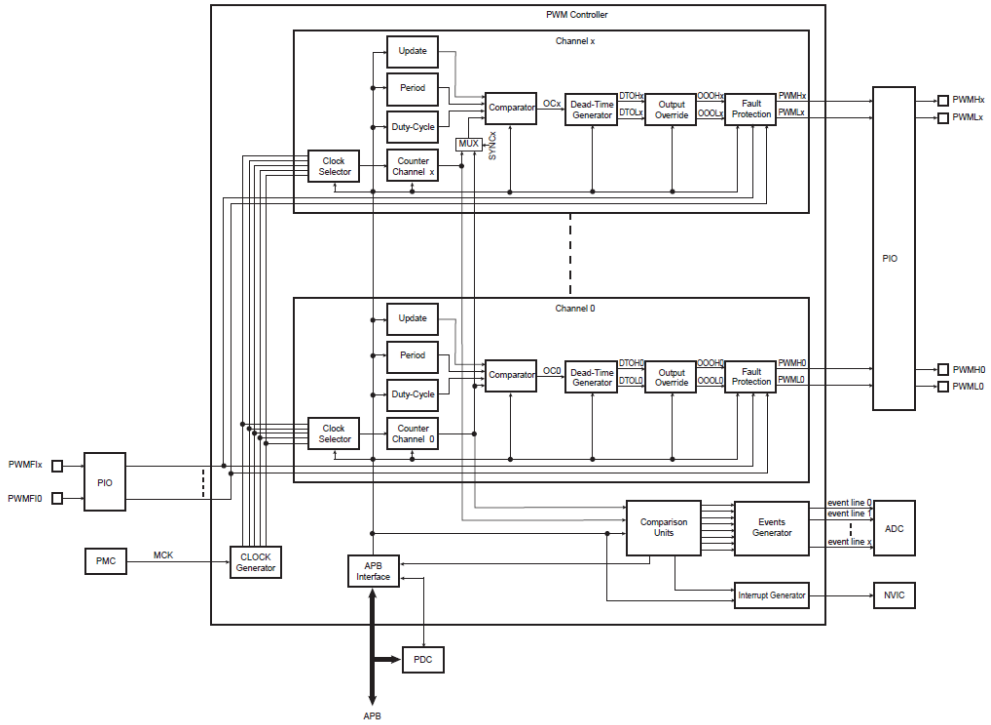


[1] Please see reference slide for more information

- The circuit above shows where S1 and S2 PWM signals feed into the MOSFETs. The reason why S1 and S2 are complementary is to prevent both MOSFETs from being ON at the same time. Since theoretically this project can be use for transmitting HIGH voltage protecting the MOSFETs is the main priority of the Gate driver and Microcontroller this is why a dead time is required for all four signals.

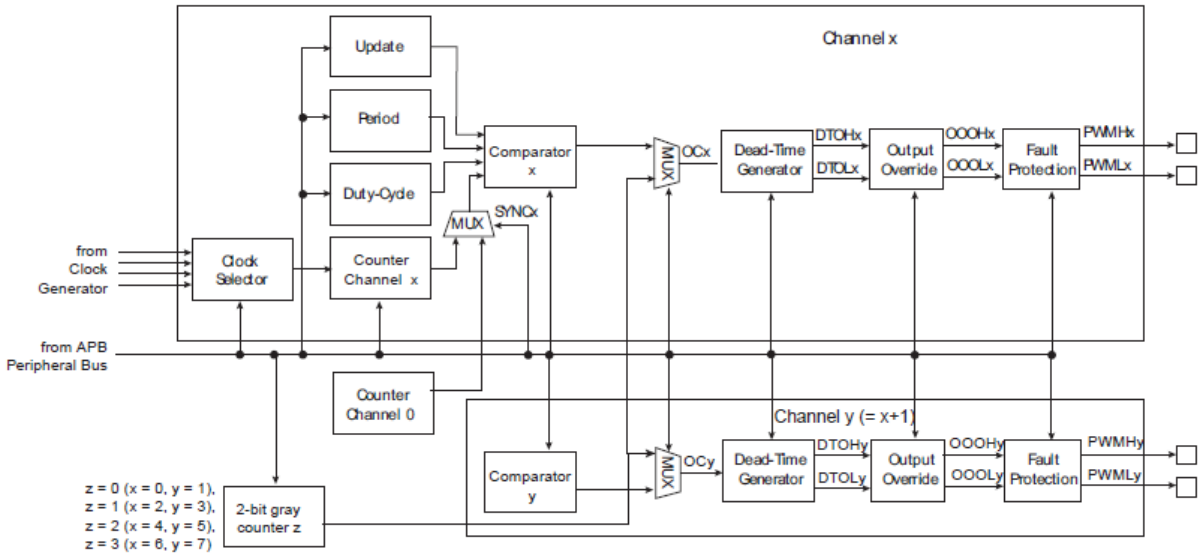
PWM REGISTER BACKGROUND

Figure 38-1. Pulse Width Modulation Controller Block Diagram



- The PWM registers for the Arduino Due can be set to use both the PWML and PWMH as inputs or outputs. Refer to the datasheet for more information.

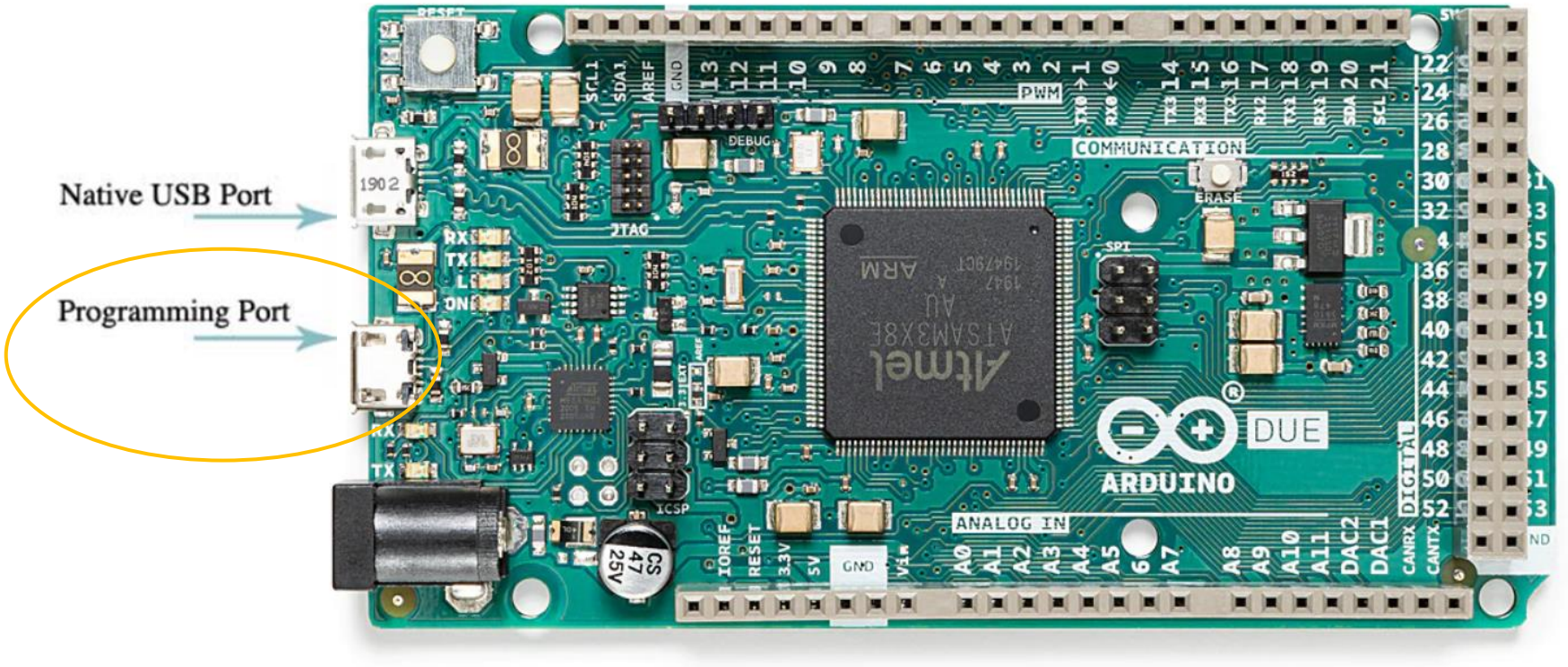
Figure 38-3. Functional View of the Channel Block Diagram



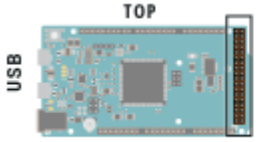
- The block diagram above shows a high-level view of the difference components that are built into the SAM3X8E which gives the user the ability to control all the following parameters by setting the correct register bits.

USB CONNECTIONS

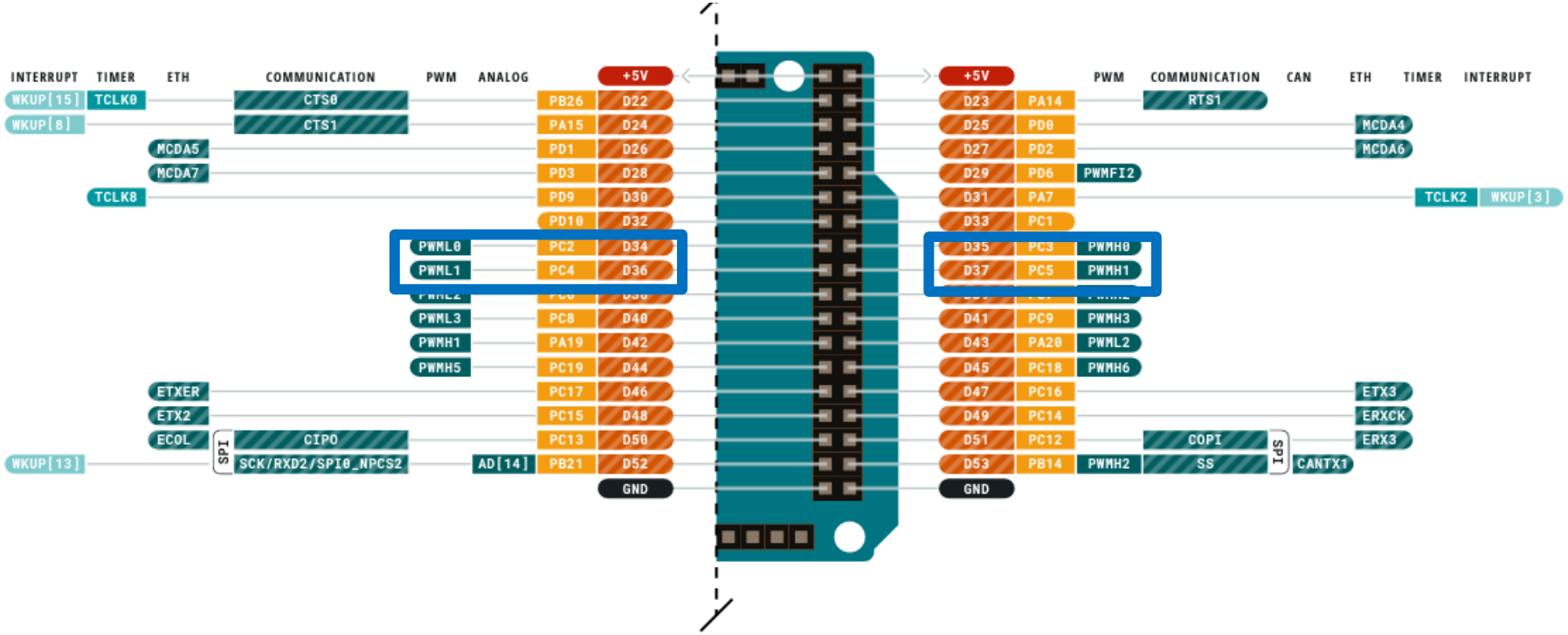
- For this project we will be using the **Programming Port** to access the SAM3X hardware UART.



ARDUINO BOARD PWM PIN CONNECTIONS

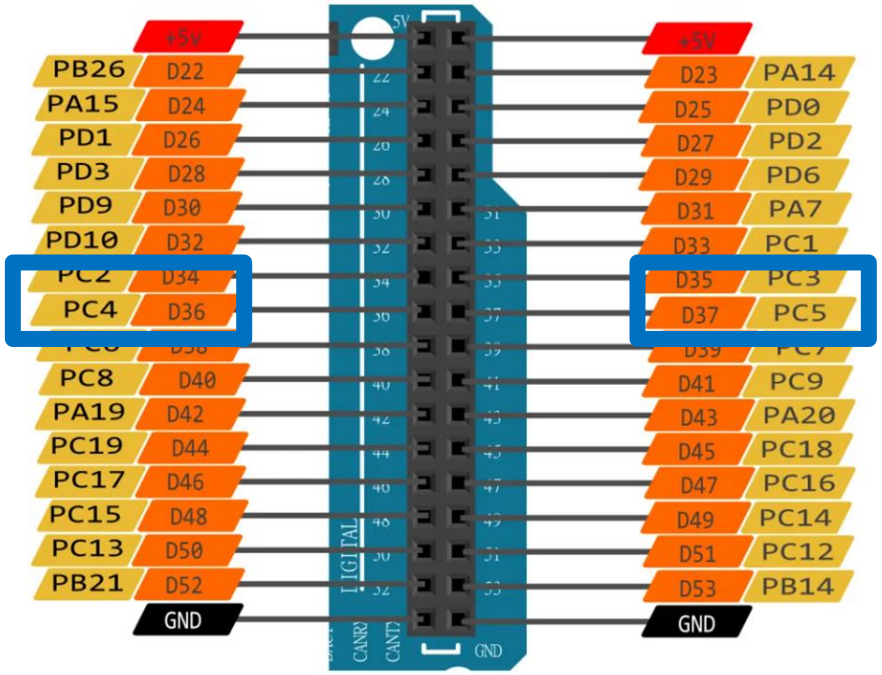


Digital pins D22-D53

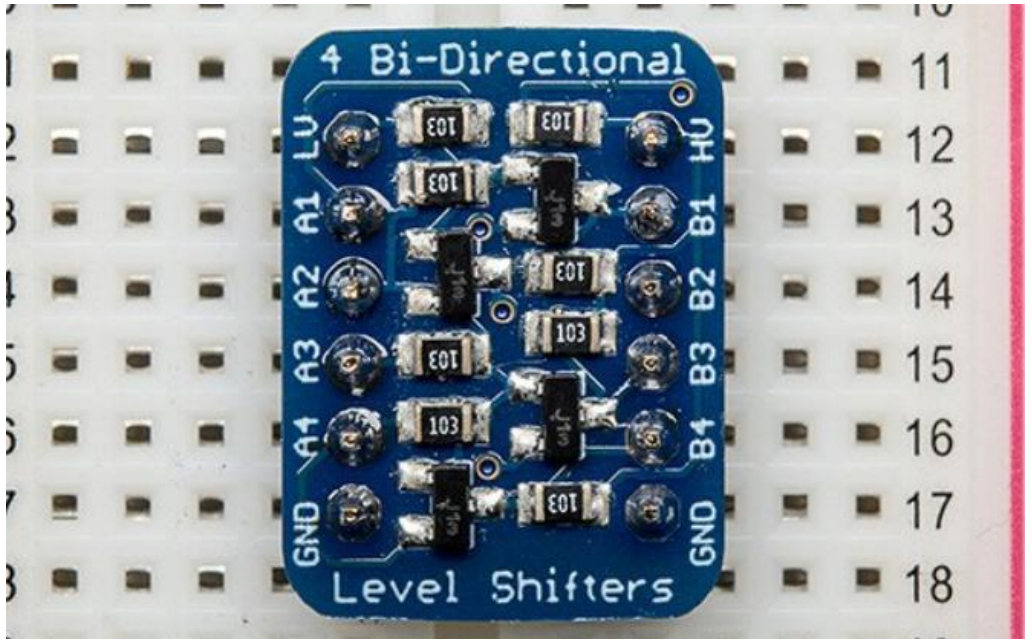


- For this project we will be using PWML0 (D34) & PWML1 (D36) as complementary pairs. As well as PWMH0 (D35) & PWMH1 (D37) as complementary pairs.

PWM CHANNEL TO LEVEL SHIFTER



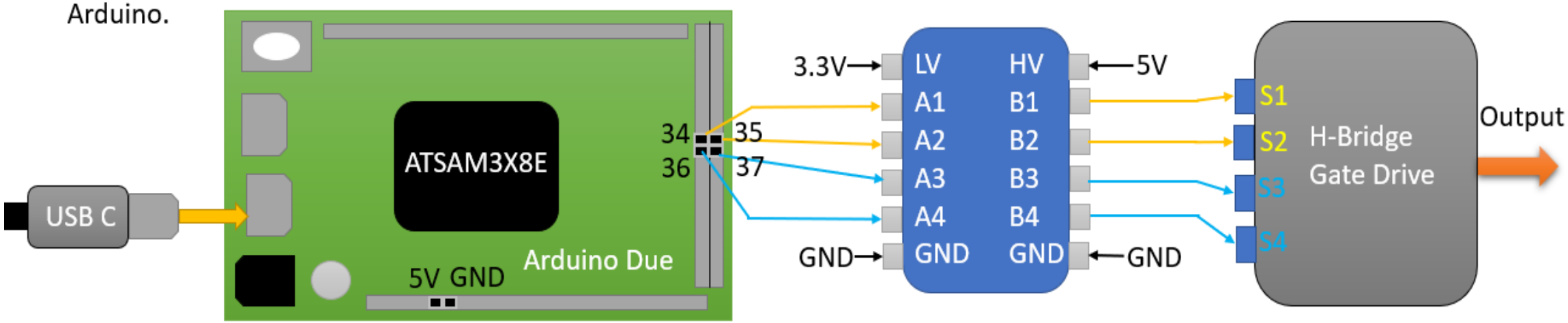
- For this project we will be using D34, D36, D35 & D37.



- The level shifter consist of 4 input channels at 3.3V and outputs 4 5V signals. The microcontroller can provide both the 3.3V and 5V required for LV & HV, respectively.

DESIGN BLOCK DIAGRAM

- Insert USB to Programming Port on the Arduino.
- Microcontroller (Pin 34 & 35 are complementary PWM signals. Pin 36 & 37 are complementary PWM signals.)
- 3.3V to 5V level Shifter.
- Full-Bridge Inverter (MOSFETs).



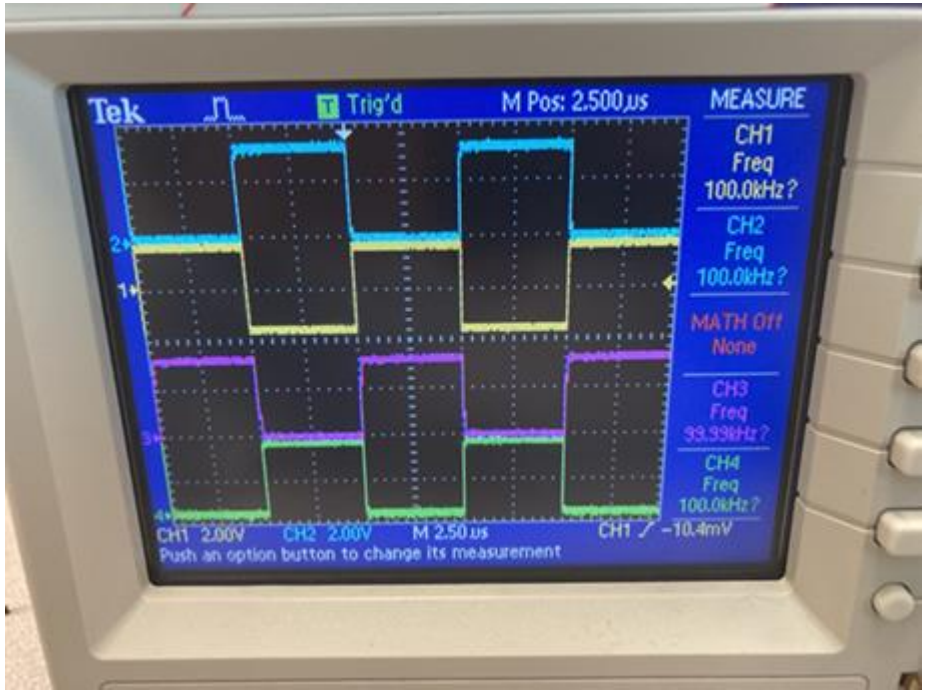
SERIAL TERMINAL FOR CONTROLLING PWM SIGNALS

COM6

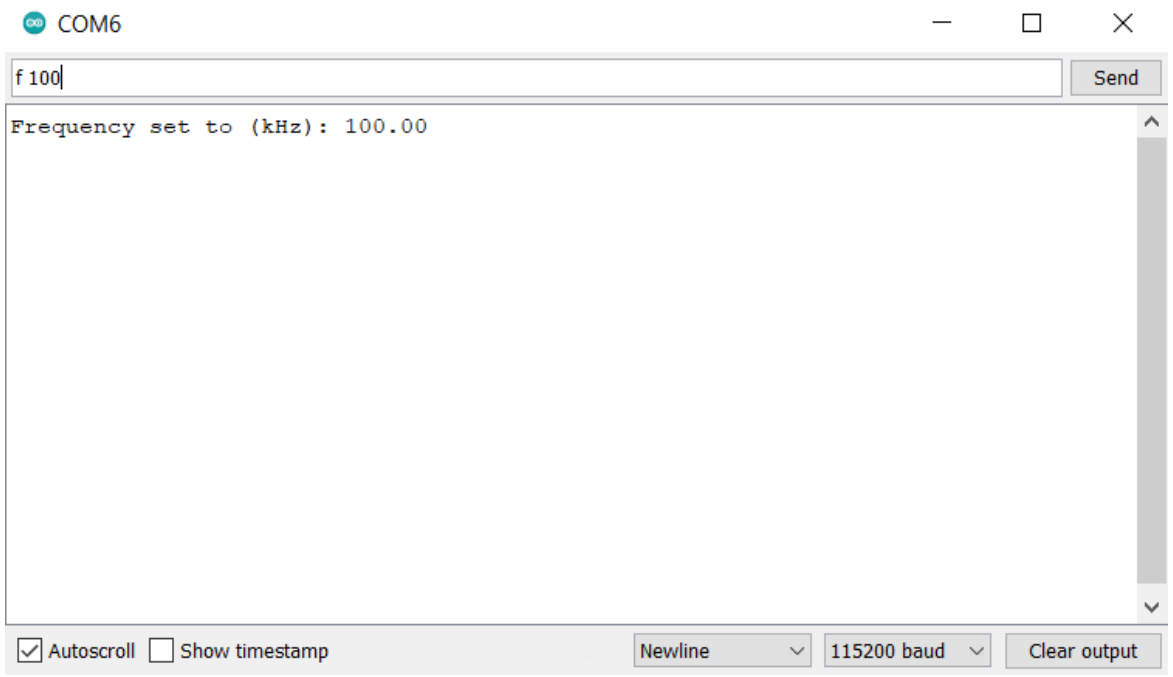
Send

```
(f) for Frequency in kHz
(d) for Duty Cycle in degrees (5-95)
(p) for Phase Shift in degrees (0-175)
(a) for Dead time insertion on Channel 0 (0-100) ns
(b) for Dead Time insertion on Channel 1 (0-100) ns
```

Autoscroll Show timestamp Newline 115200 baud Clear output

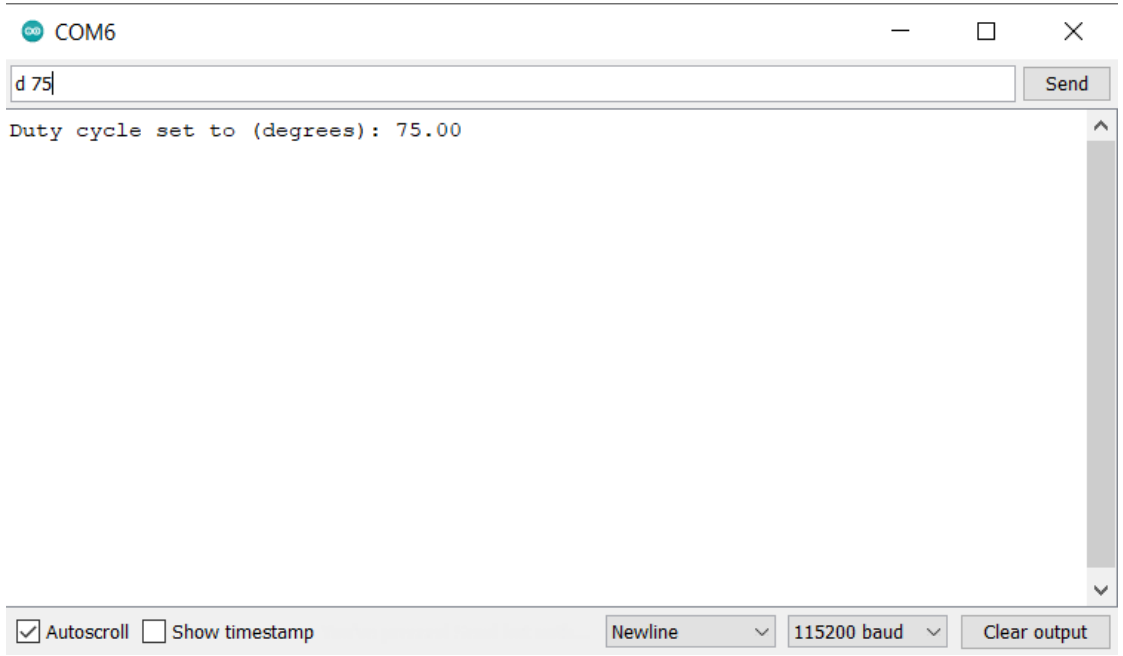


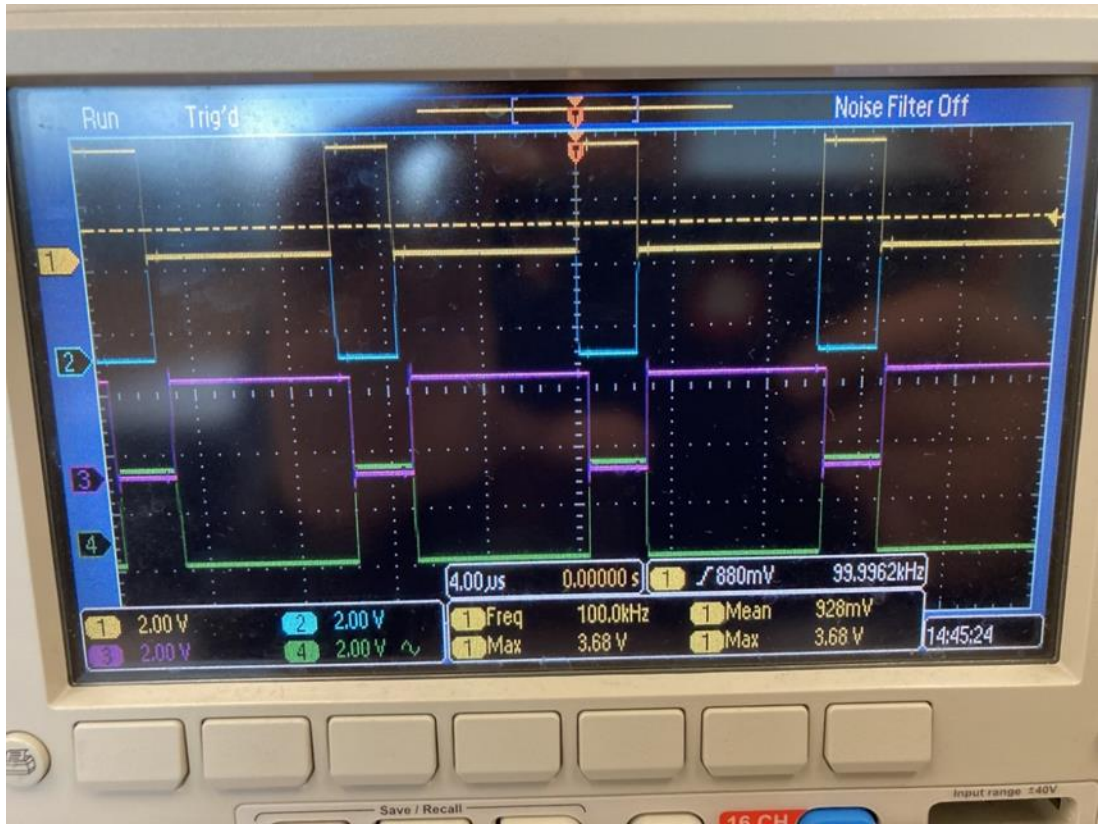
SERIAL TERMINAL: FREQUENCY ADJUSTMENT (KHZ)



- To set or change the frequency use type letter f followed by any number ranging from 5kHz-500kHz.

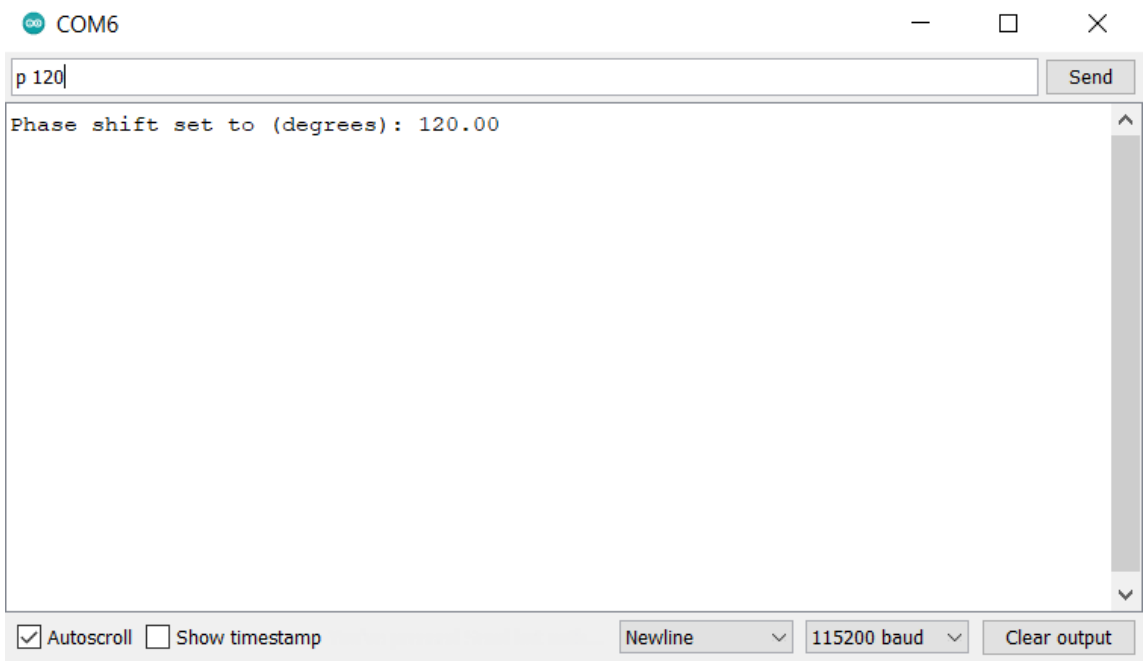
SERIAL TERMINAL: DUTY CYCLE MODE (DEGREES)

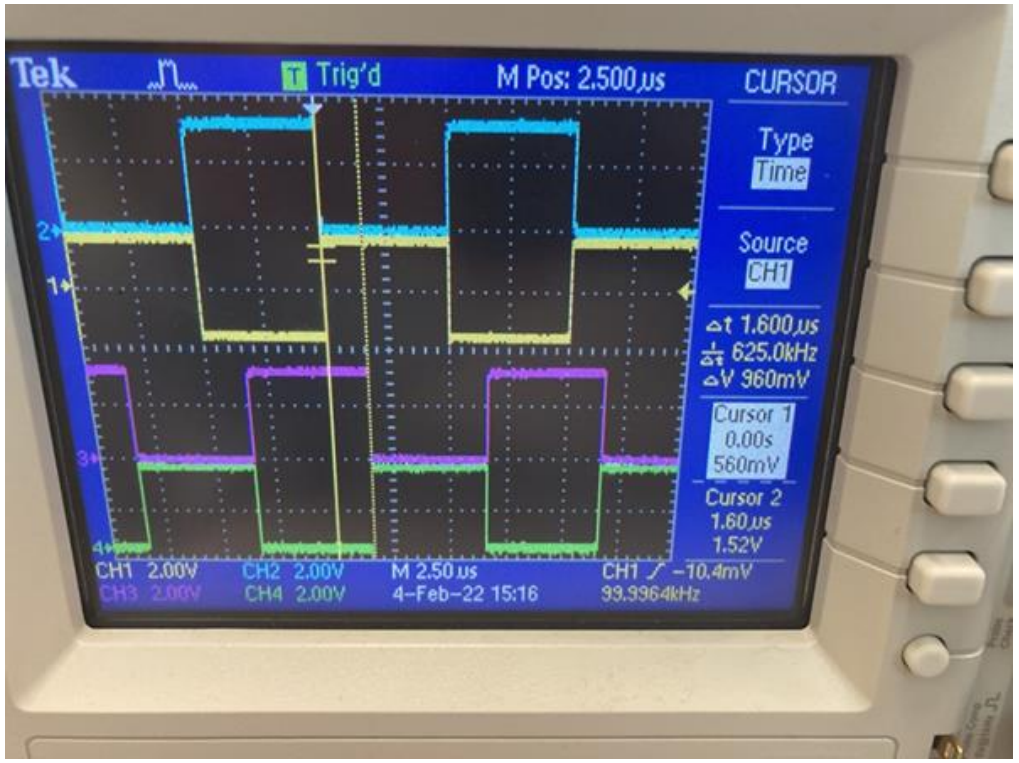




- To change the duty cycle type letter d followed by any number ranging from 0-100 degrees.

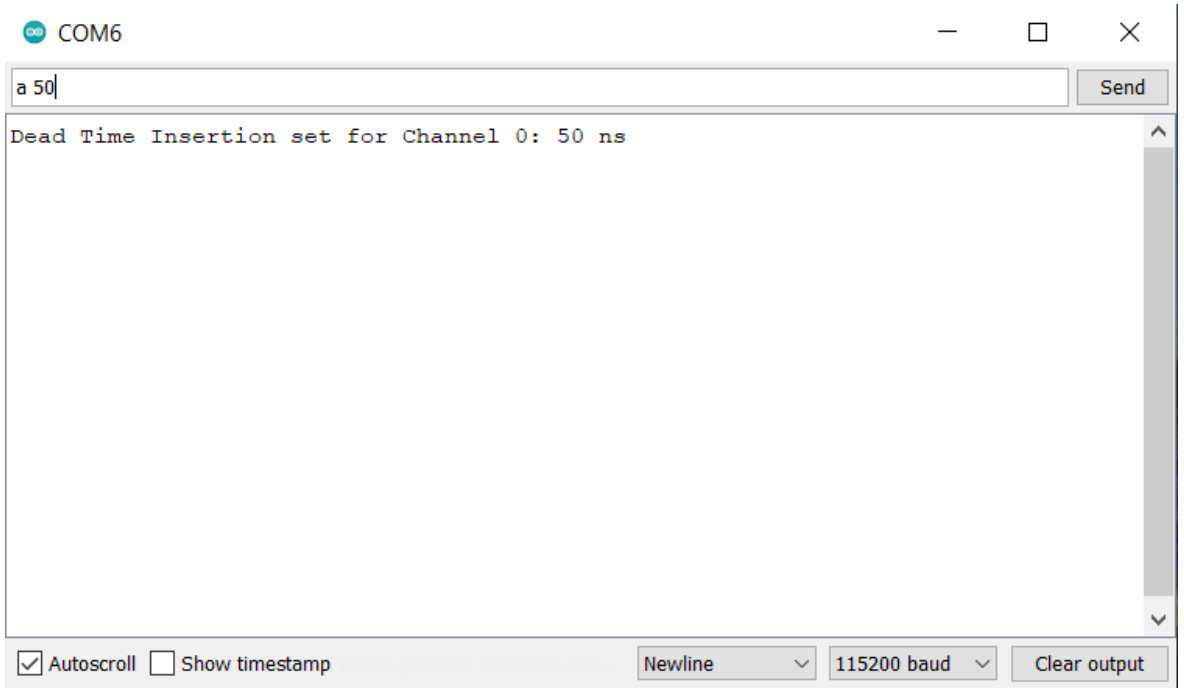
SERIAL TERMINAL: PHASE SHIFT MODE (DEGREES)

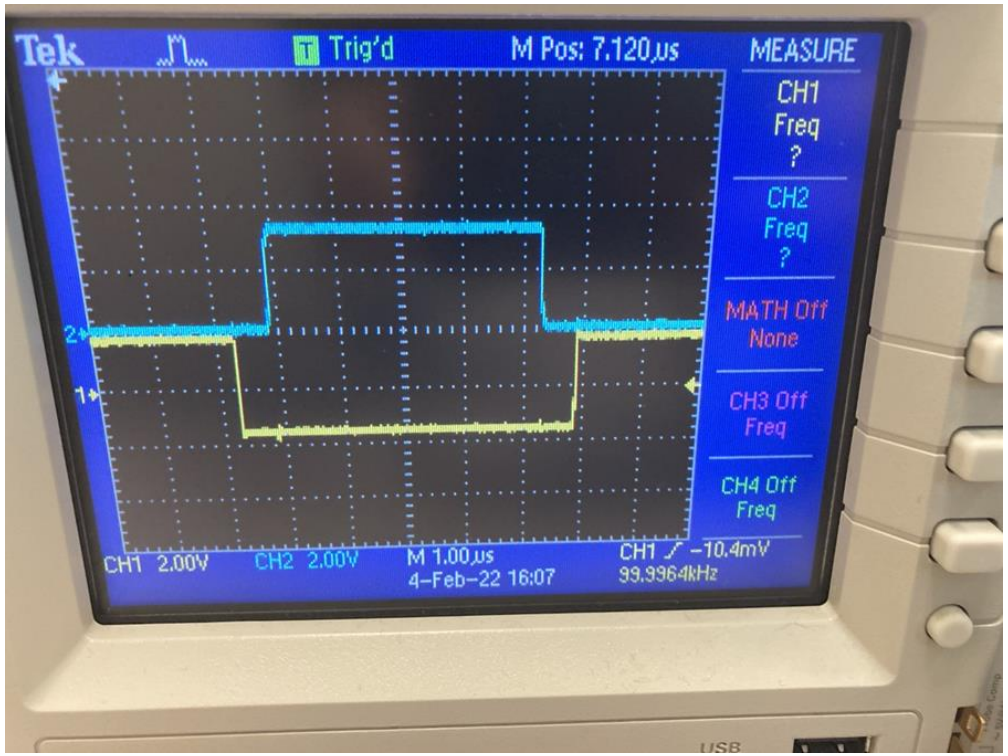




- To control the phase shift type letter p followed by any number from 0-180 degrees.

SERIAL TERMINAL: DEAD-TIME INSERTION (NANO-SECONDS)

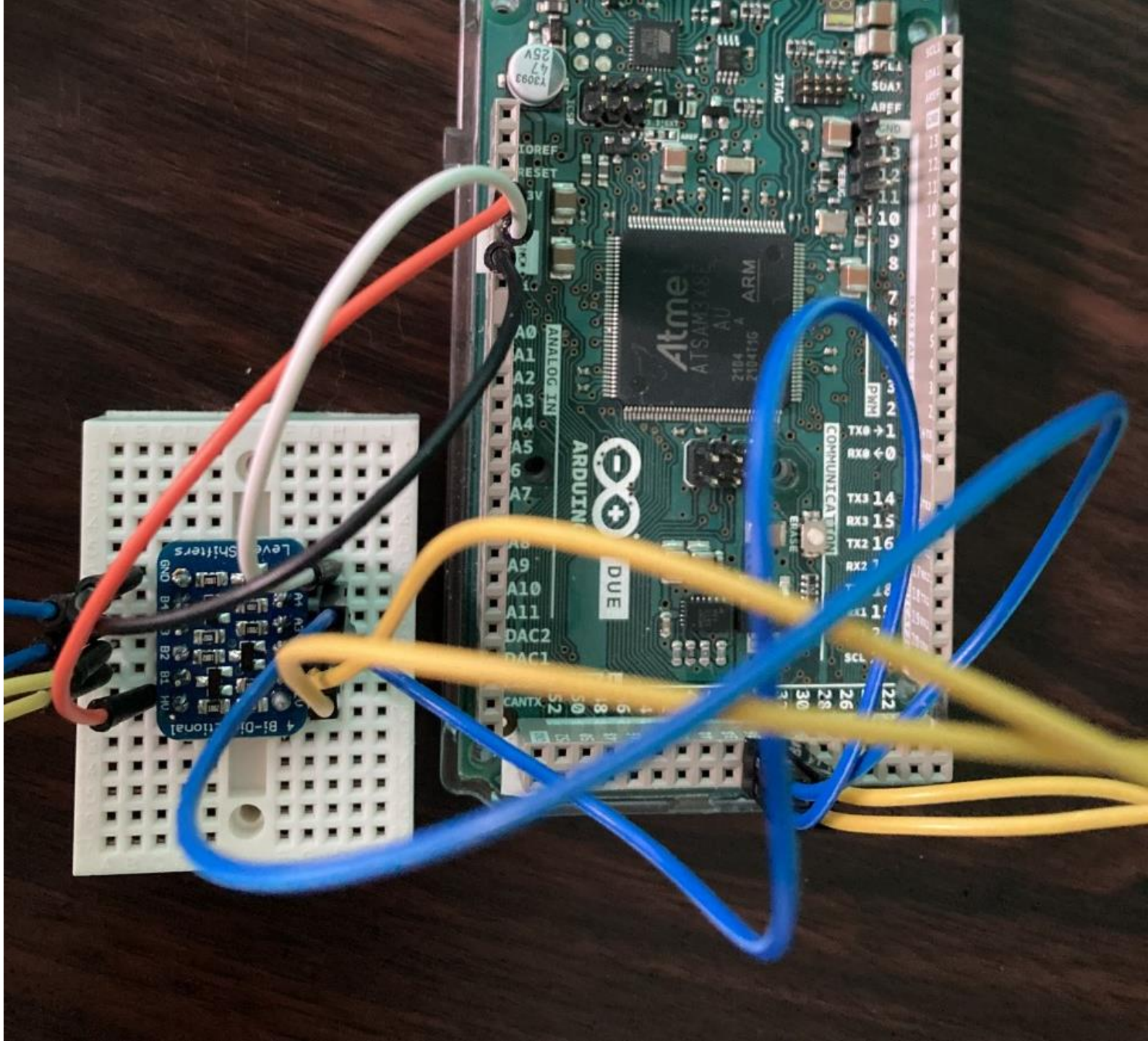




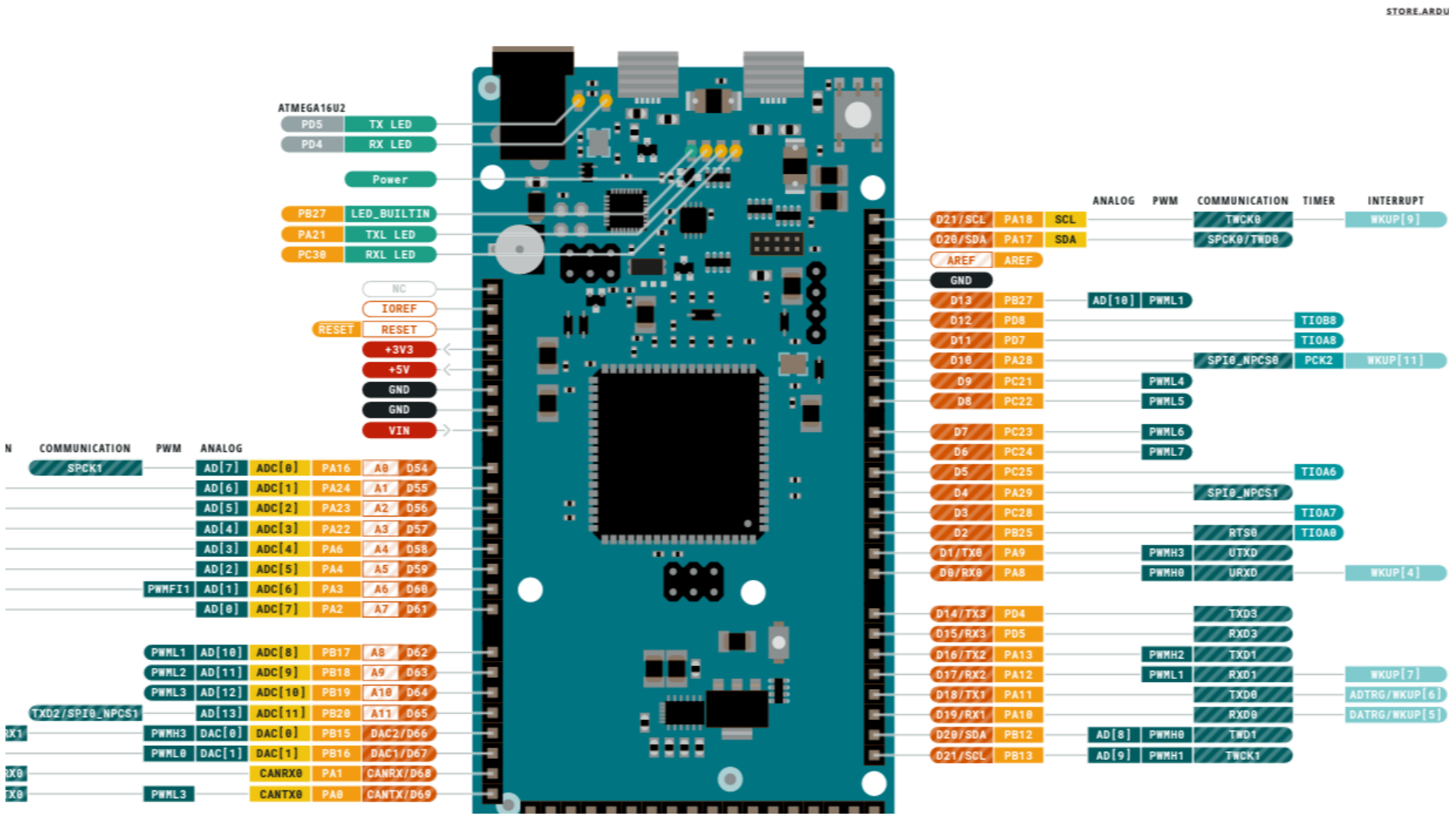
- To change the dead-time for complementary pair 1 type the letter a followed by any value in the range of 0-350. For Complementary pair 2 type b.

FINAL DESIGN

- To the right you can see all the connections to get the PWM signal to a 5V output.

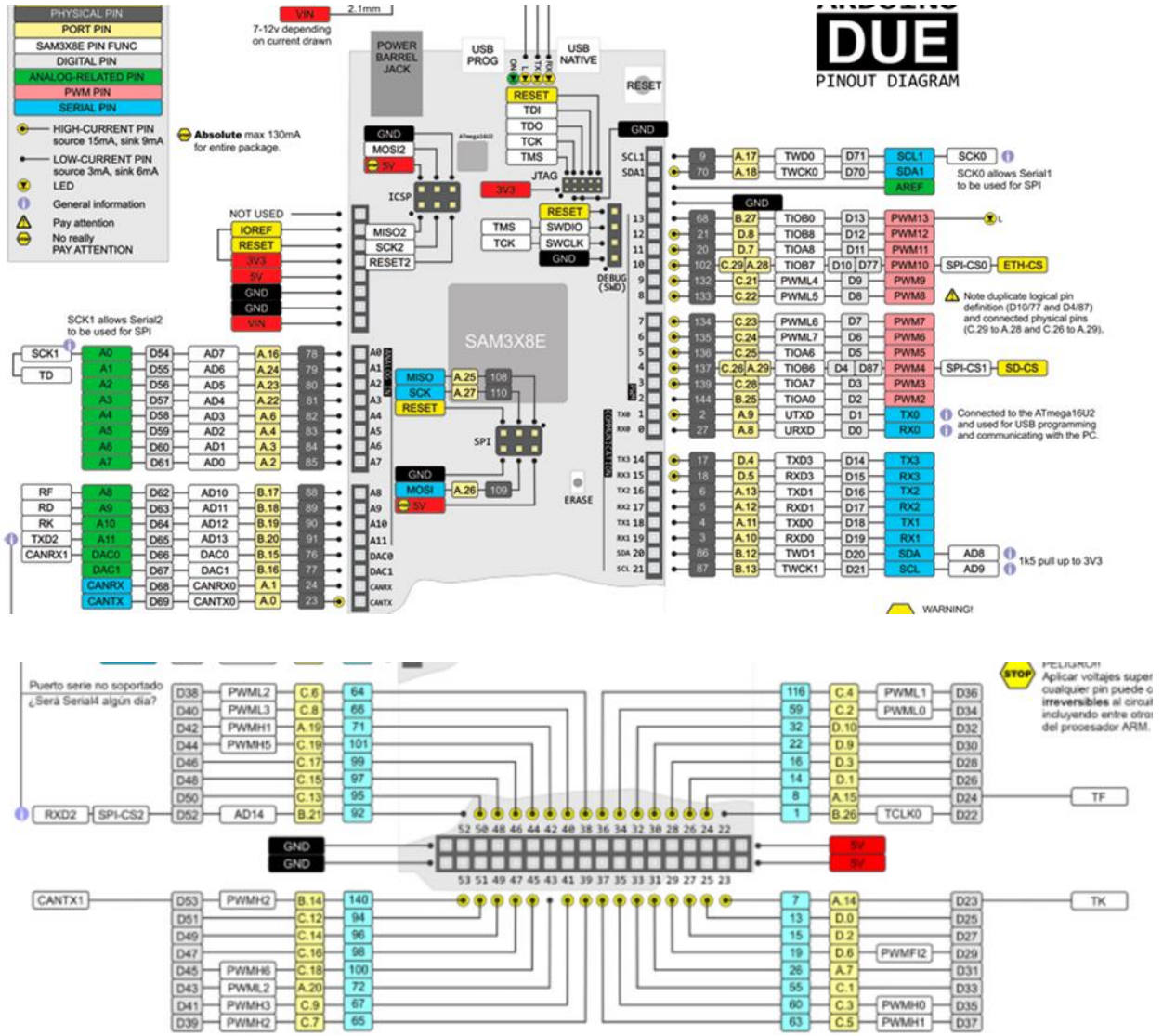


Additional Resource for Arduino Due PIN-OUT



ADDITIONAL PIN NOTATIONS

- When adding & enabling more PWM channels please refer to the PIN-OUT names and location on the diagram to the right. Use the SAM2X8E PIN FUNC to find the PWMH#/PWML# pairs.



REFERENCES

- [1] H. Hu, T. Cai, S. Duan, X. Zhang, J. Niu and H. Feng, "An Optimal Variable Frequency Phase Shift Control Strategy for ZVS Operation Within Wide Power Range in IPT Systems," in IEEE Transactions on Power Electronics, vol. 35, no. 5, pp. 5517-5530, May 2020, doi: 10.1109/TPEL.2019.2947092.