

Lab 10F Report

Requirements Document

1. Overview

1.1. Objectives: Why are we doing this project? What is the purpose?

The objectives of this project are to design, build and test a brushed DC motor controller. The motor should spin at a constant speed and the operator can specify the desired set point. Educationally, students are learning how to interface a DC motor, how to measure speed using input capture, and how to implement a digital controller running in the background.

1.2. Process: How will the project be developed?

The project will be developed using the TM4C LaunchPad and the Blynk App. The Blynk App will specify the desired speed of the motor as well as the “K” terms for the Proportional and Integral components of the PI controller. The system will be built on a solderless breadboard and run on the usual USB power. A hardware/software interface will be designed that allows software to control the DC motor. There will be at five hardware/software modules: ESP8266 WiFi unit, tachometer input, motor controller output, LCD output. There will also be two software only components that interface the motor controller to the Blynk App: TM4C \Rightarrow Blynk and the Blynk \Rightarrow TM4C. The process will be to design and test each module independently from the other modules. After each module is tested, the system will be built and tested.

1.3. Roles and Responsibilities: Who will do what? Who are the clients?

EE445L students are the engineers and the TA is the client. Student A will build and test the sensor system. Student B will build the actuator and Blynk input. Both students will work on the controller. *(note to students: you are expected to make minor modifications to this document in order to clarify exactly what you plan to build. Students are allowed to divide responsibilities of the project however they wish, but, at the time of demonstration, both students are expected to understand all aspects of the design.)*

1.4. Interactions with Existing Systems: How will it fit in?

The system will use the microcontroller board, a solderless breadboard, and the DC motor shown above in Figure 10.2. The wiring connector for the DC motor is described in the PCB Artist file **Lab10E_Artist.sch**. It will be powered using the USB cable. **You must use a +5V power from the lab bench, but please do not power the motor with a voltage above +5V. Do not connect this bench supply to Vbus (LaunchPad +5V). However, you must have a common ground.**

1.5. Terminology: Define terms used in the document.

Torque - The tendency of a force to rotate an object about an axis, fulcrum, or pivot. Just as a force is a push or a pull, a torque can be thought of as a twist to an object. It is units of Newton-meters or $N \cdot m$

PWM – Pulse width modulation is used to control the power to an element by sending digital pulses of varying duty cycles at high frequency so that the objects average power for one cycle is reflected.

Hysteresis - the dependence of the state of a system on its history. For example, a magnet may have more than one possible magnetic moment in a given magnetic field, depending on how the field changed in the past.

Proportional-Integral Controller - A variation of Proportional Integral Derivative (PID) control is to use only the proportional and integral terms as PI control. The PI controller is the most popular variation, even more than full PID controllers. The value of the controller output $u(t)$ is fed into the system as the manipulated variable input.

Back EMF - A motor has coils turning inside magnetic fields, and a coil turning inside a magnetic field induces an emf. This emf, known as the back emf, acts against the applied voltage that's causing the motor to spin in the first place, and reduces the current flowing through the coils of the motor.

Time Constant - a time that represents the speed with which a particular system can respond to change, typically equal to the time taken for a specified parameter to vary

1.6. Security: How will intellectual property be managed?

The system may include software from TivaWare and from the book. No software written for this project may be transmitted, viewed, or communicated with any other EE445L student past, present, or future (other than the lab partner of course). **It is the responsibility of each team to keep its EE445L lab solutions secure.**

2. Function Description

2.1. Functionality: What will the system do precisely?

The palette for the Blynk App (Fig 10.4) will have the following 8 virtual pin components:

- Slider to control the desired motor speed
- Two sliders to control the “K” multiplier/divider for the Proportional term
- Two sliders to control the “K” multiplier/divider for the Integral term
- Slider to control the “Loop Response” of the PI controller (Tau)
- RPS Gauge that graphically shows the actual speed of the motor
- RPS Value display

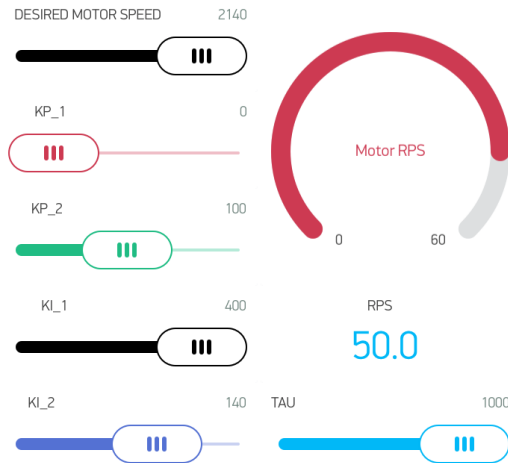


Figure 10.4 Blynk App palette (feel free to remove the Motor RPS so you'll not have to buy tokens) You may also have to delete the Lab 4 Blynk app

The PI control equation should be in this form:

```

P = (Kp1*E)/Kp2;           // Proportional term
I = I+(Ki1*E)/Ki2;         // SUM(KiDt)
U = P + I                  // Sum Proportional plus Integral terms
PWM0A_Duty(U);             // output

```

The motor speed should start out at zero RPS. Once the desired motor speed slider is adjusted the motor should start. (Note to students: feel free to change how the set point is established, and feel free to increase or decrease the maximum speed in accordance to how it actually works.)

Both the desired and actual speeds should be plotted on the color LCD as a function of time similar to Figure 10.10. The actual speed should also be graphically shown on the Blynk App. (note to students: feel free to specify exactly how the data is displayed. For example, you could but do not have to add numerical outputs).

2.2. Scope: List the phases and what will be delivered in each phase.

Phase 1 is the preparation; phase 2 is the demonstration; and phase 3 is the lab report. Details can be found in the lab manual.

2.3. Prototypes: How will intermediate progress be demonstrated?

A prototype system running on the LaunchPad and solderless breadboard will be demonstrated. Progress will be judged by the preparation, demonstration and lab report.

2.4. Performance: Define the measures and describe how they will be determined.

The system will be judged by three qualitative measures. First, the software modules must be easy to understand and well-organized. Second, the system must employ an integral controller running in the background. There should be a clear and obvious abstraction, separating the state estimator, user interface, the controller and the actuator output. Backward jumps in the ISR are not allowed. Third, all software will be judged according to style guidelines. Software must follow the style described in Section 3.3 of the book (note to students: you may edit this sentence to define a different style format). There are three quantitative measures. First, the average speed error at a desired speed of 60 RPS will be measured. The average error

should be less than 5 RPS. Second, the step response is the time it takes for the new speed to hit 60 RPS after the set point is changed from 40 to 60 RPS. Third, you will measure power supply current to run the system. There is no particular need to minimize controller error, step response, or system current in this system.

2.5. Usability: Describe the interfaces. Be quantitative if possible. Describe the how the Blynk App will control the motor.

The Blynk app will communicate to the TM4C through an ESP8266 module wired to UART5 on the TM4C. The Blynk app will send a string of data that will have a designated virtual pin number and the value associated with the pin that was changed or updated on the Blynk app. The TM4C will parse the incoming UART5 data and associate the virtual pin number to a variable or function and update that function or variable with the value sent from the Blynk app.

2.6. Safety: Explain any safety requirements and how they will be measured.

Figure 10.3 shows that under a no-load condition, the motor current will be less than 100 mA. However, under heavy friction this current could be 5 to 10 times higher. Therefore, please run the motors unloaded. Connecting or disconnecting wires on the protoboard while power is applied will damage the microcontroller. Operating the circuit without a snubber diode will also damage the microcontroller.

3. Deliverables

3.1. Reports: How will the system be described?

A lab report described below is due by the due date listed in the syllabus. This report includes the final requirements document.

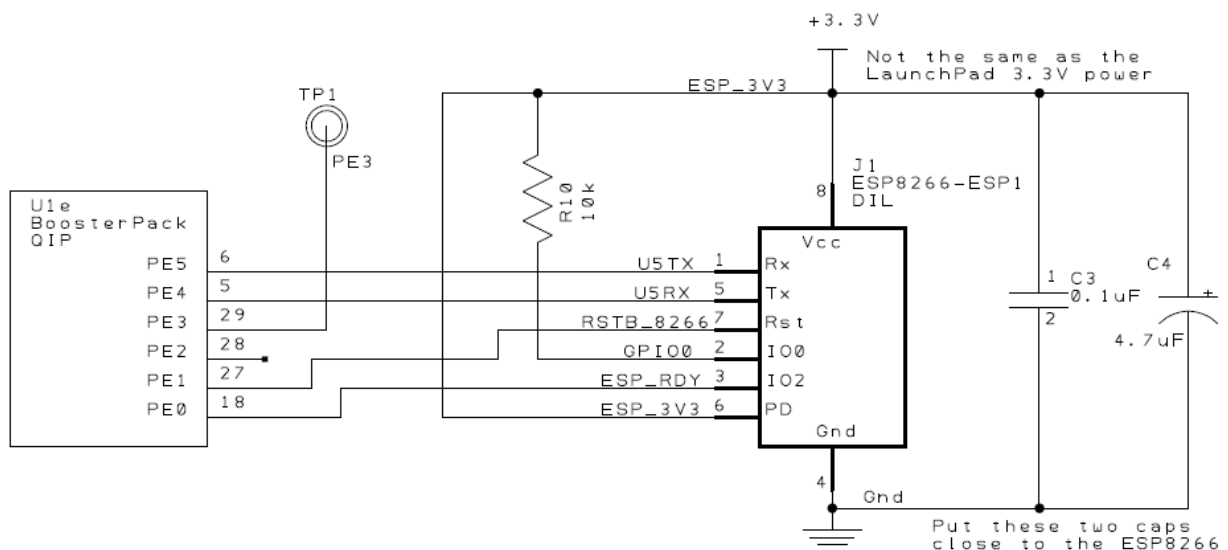
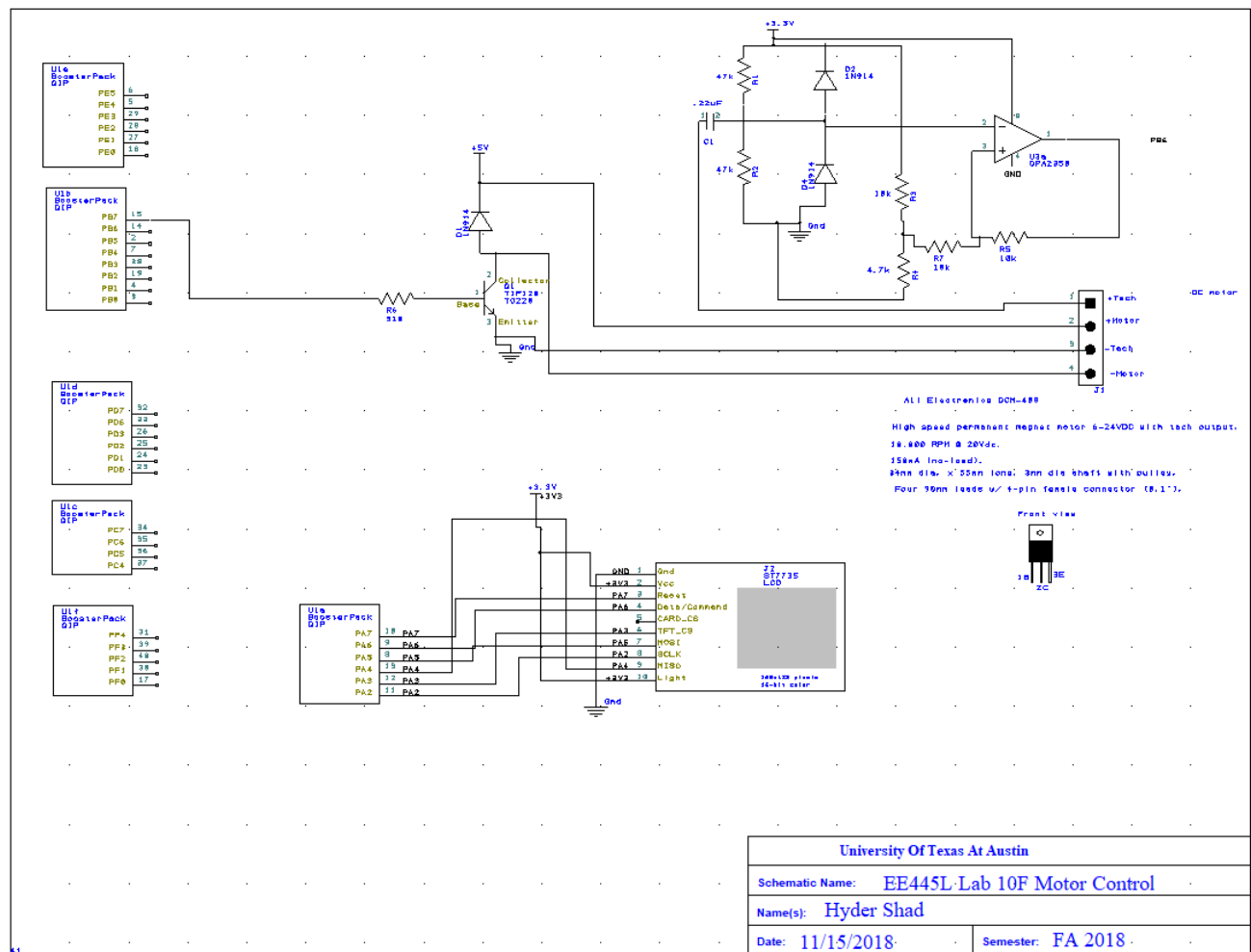
3.2. Audits: How will the clients evaluate progress?

The preparation is due at the beginning of the lab period on the date listed in the syllabus.

3.3. Outcomes: What are the deliverables? How do we know when it is done?

There are three deliverables: preparation, demonstration, and report. (Note to students: you should remove all notes to students in your final requirements document).

The circuit for Lab 10F was designed in a PCB Artist file and the schematic is provided below:



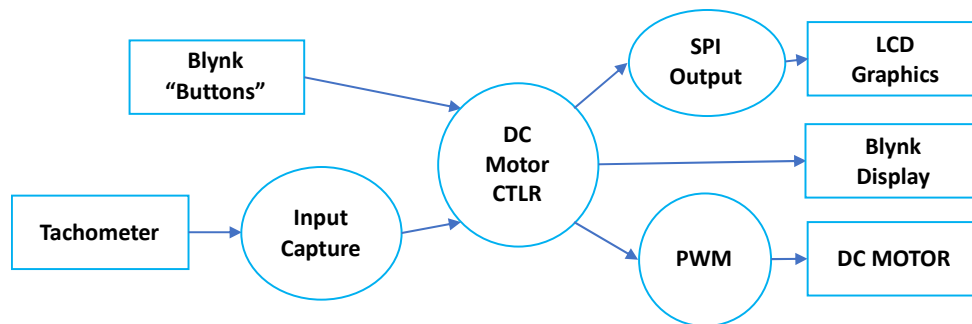
Hardware interfaces include the following:

- DC motor voltage input controlled via a TP120 connected to the PWM output on the TM4C
- LCD screen showing RPS plot, current RPS, and Set/Target RPS
- Tachometer analog sinusoidal wave to digital pulse output using OPA2530 amplifier circuit. Pulse output from op-amp is connected to edge sensitive input on TM4C.
- ESP8266 module powered by an LM2937 3.3v regulator and connected to the UART5 pins on the TM4C

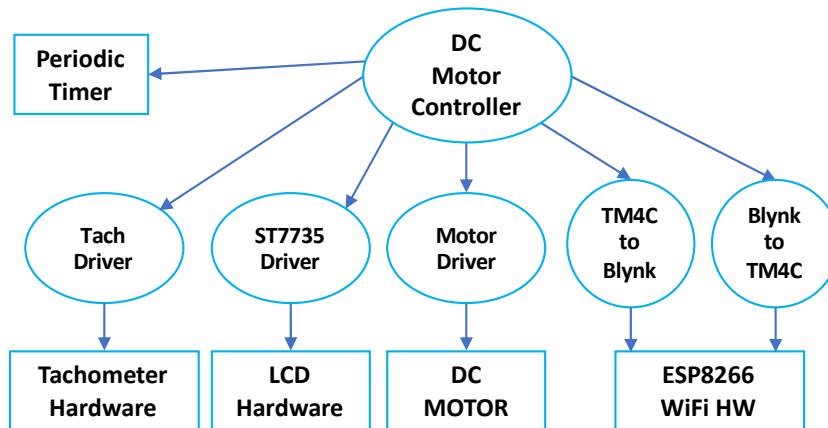
The hardware schematics were included in the lab submission

Software Design

The software modules were based of the provided data flow charts and call graphs in the lab documentation as shown below:



Data flow chart



Function call graph

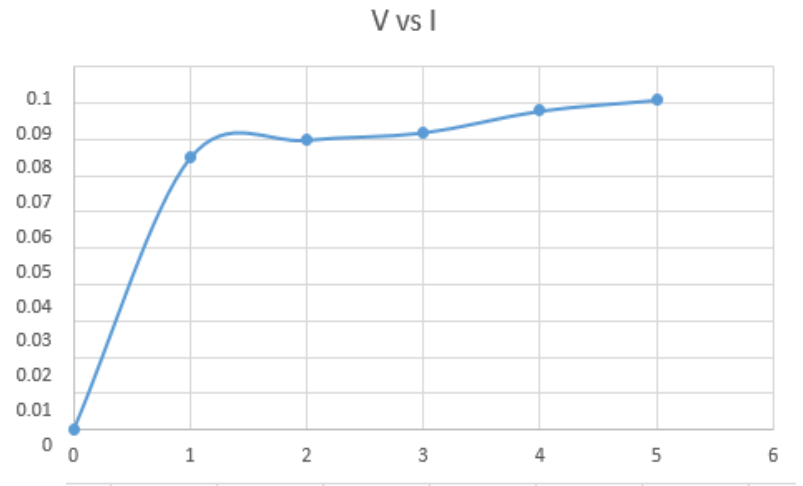
The software modules were included in the lab submission.

Measurement Data

Procedure 1, Give the voltage, current, and resistance measurements (No Load):

Applied Voltage (Volts)	Coil Current (Amps)
0	0
1	0.084
2	0.087
3	0.092
4	0.095
5	0.098

Resistance (No Load) = 53.8 Ohms

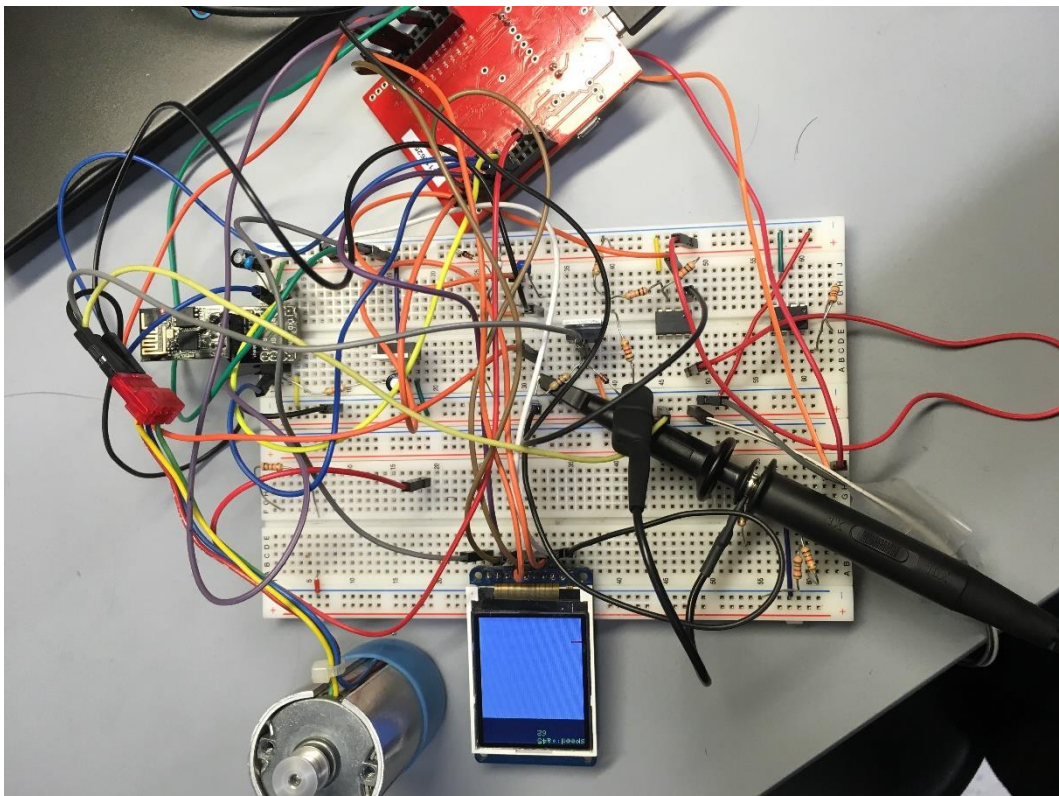


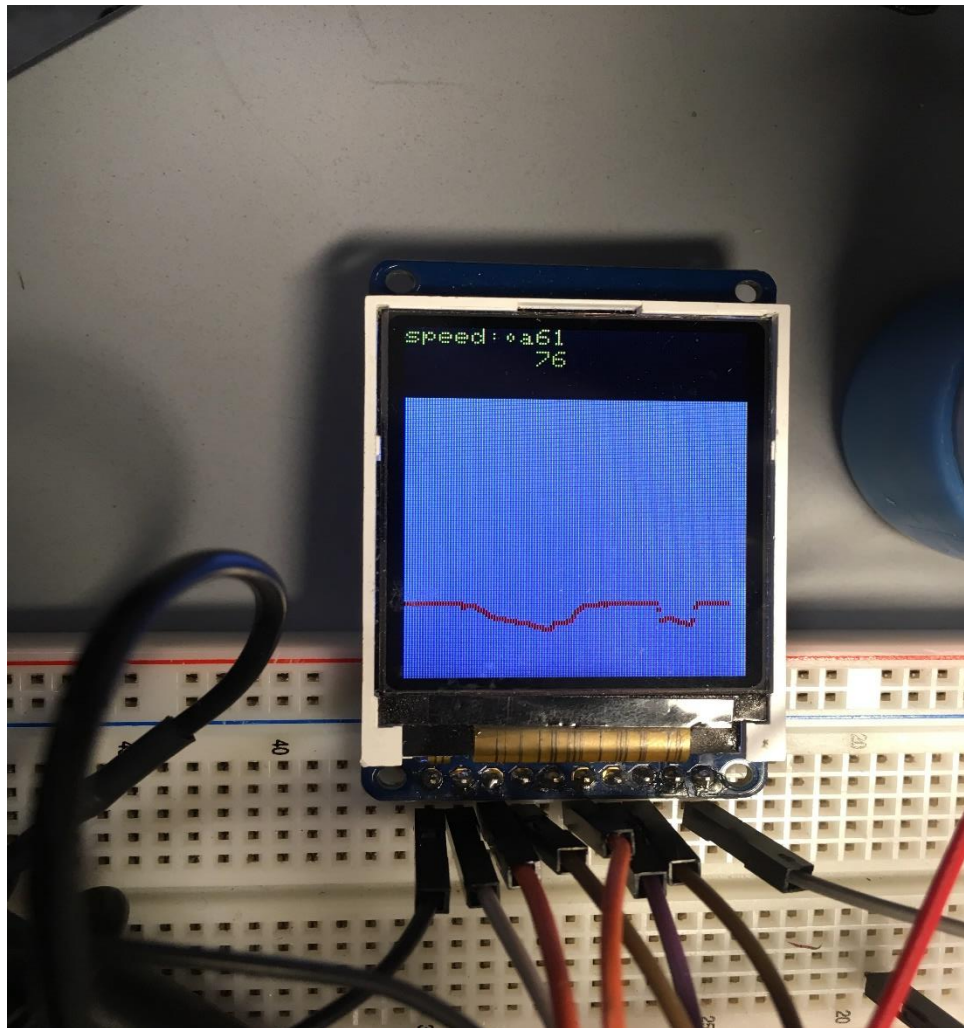
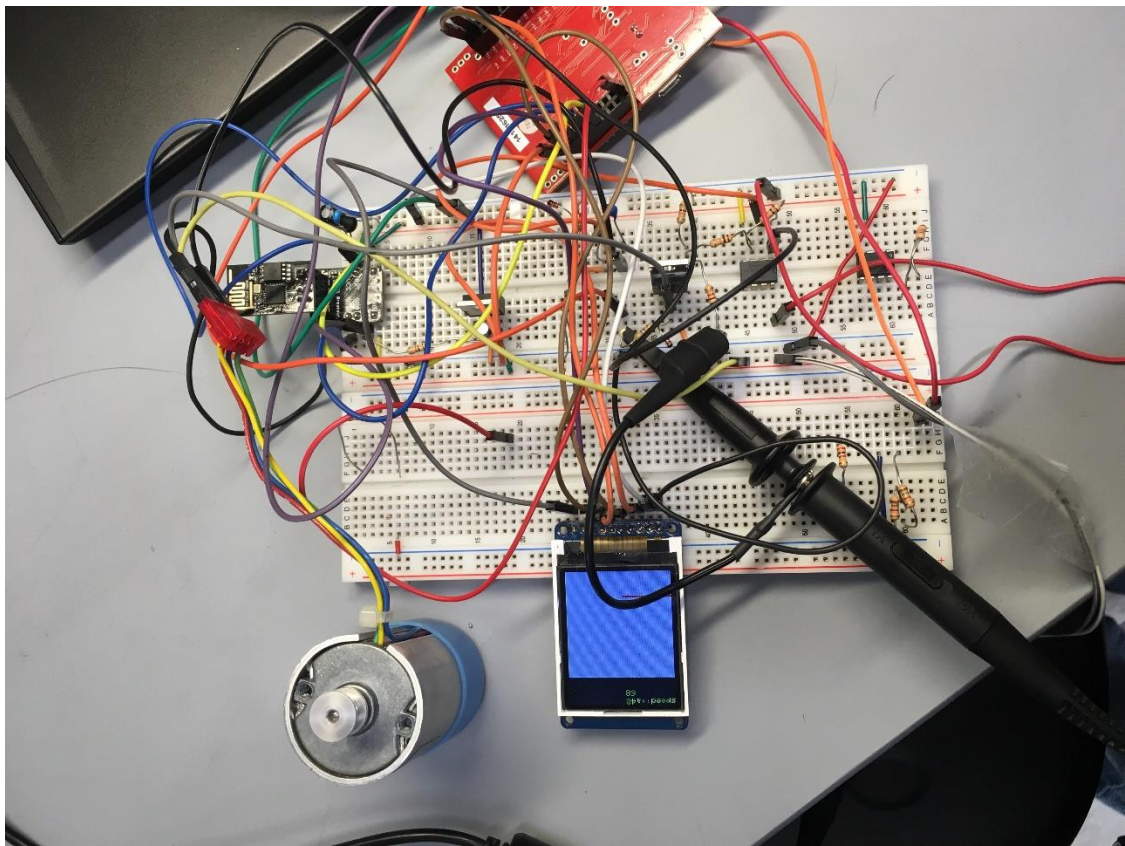
Procedure 2, I_{BE} and I_{CE} while spinning (measured using multimeter):

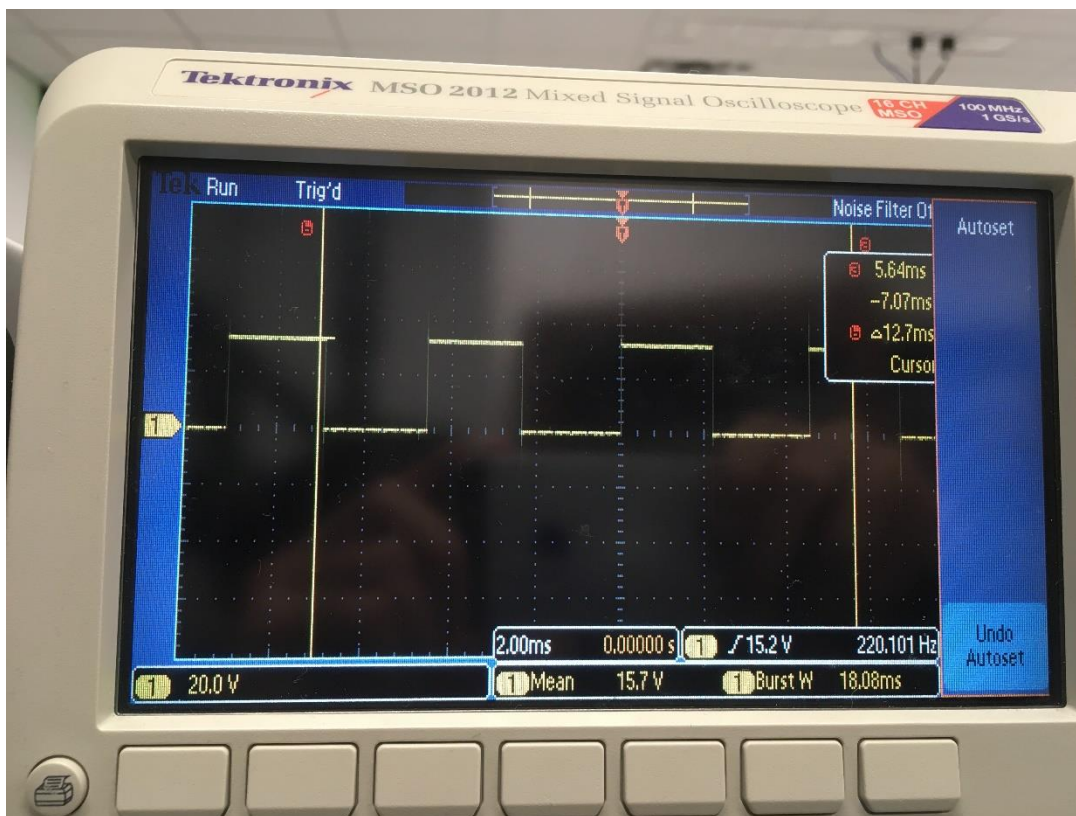
$I_{BE} = 0.00395$ A

$I_{CE} = 0.088$ A

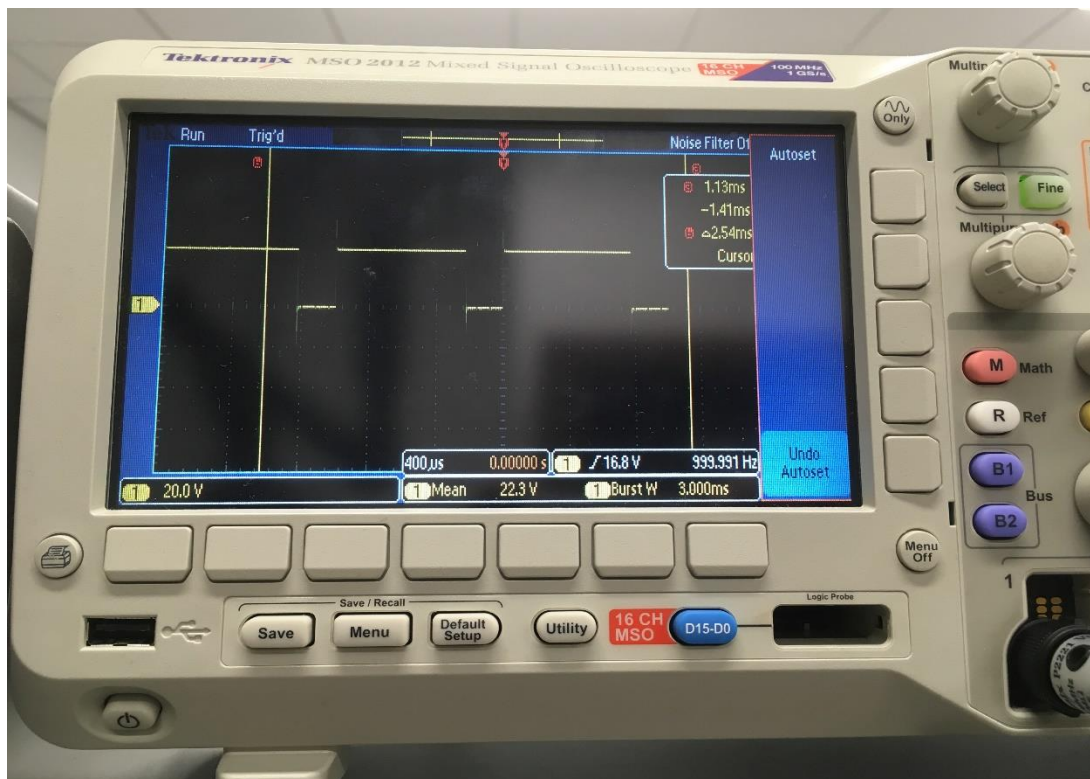
Procedure 3, Two screen shots of the hardware in operation:







Above: Op-Amp circuit pulse output in response to analog tachometer input



Above: PWM output from TM4C to TIP120 (motor control circuit)

Procedure 4, Specify the maximum time to execute once instance of the ISR:

ISR execution time for PI control handler is 2.35uS

Specify the average controller error:

Average error was shown to be approximately 0.6 RPS once at a stable speed.

Specify the approximate response time:

Approximate response time to go from 20 to 50 RPS is 2 seconds

Procedure 5, Measurements of current required to run the system, with and without the motor spinning:

Without the motor active, the TM4C and esp8266 module were the dominant power consuming components, with all hardware using approximately 0.085mA

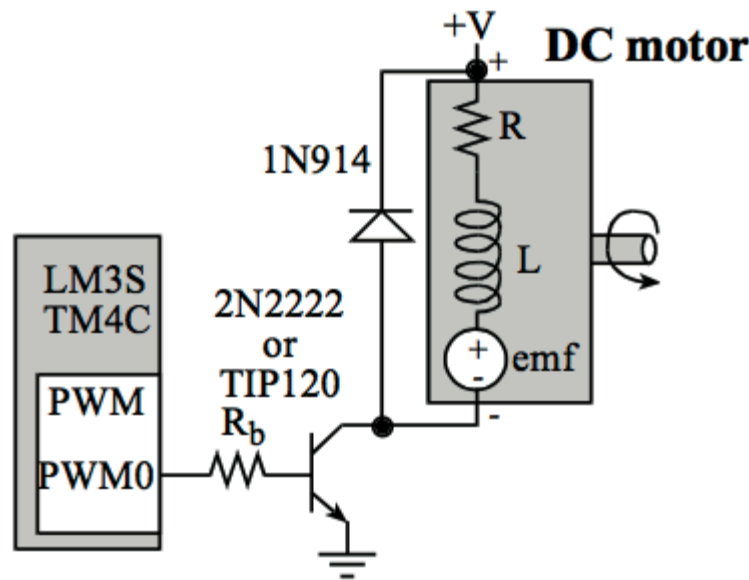


With the motor spinning at 55 RPS, all hardware used in total 0.173 A.

Analysis and Discussion

1. Torque is a twisting/rotational force, found by taking the force applied in newtons multiplied by the distance from the center of rotation the force is applied to. The units are therefore newton-meters, also commonly converted to pound-feet.

2.



A DC motor is from a circuit standpoint a resistor in series with an inductor. A load on the DC motor will cause the axle to spin slower and thus the coil will be energized for a longer period of time. This long period of time where the coil (inductor) is held at potential results in the inductor acting as a short, allowing more current to flow through the motor.

3. The TIP120 was selected over the 2N2222 because it is able to provide a higher current gain for the motor. This transistor is implemented in the motor control circuit and is switched on and off by the TM4C PWM output connected to its base pin.
4. An alternative method to control the motor would be to use a table of associated PWM duty cycles used for different RPS or an equation that related the PWM duty cycle to the motor RPS. This would reduce ISR time or eliminate the need for the PI control interrupt handler routine.
5. The motor is not always spinning at a constant rate, hence why we are implementing a PI controller to make adjustments and minimize the difference between the desired speed and actual speed. The electrical power of the motor is given in Watts by $V \cdot I$, the voltage applied to the motor multiplied by the current it draws. The mechanical power of the motor is given in the Torque of the motor multiplied by the angular velocity of the output shaft ($T \cdot \omega$), and this is measured in Watts. Both electrical and mechanical power for the motor are linear relationships, and when determining either form of power, there will be some disparity due to losses from friction or electrical resistance in which energy is dissipated as heat.