Для выполнения домашнего задания Вам потребуется:

1) Скачать и установить Proteus 8.7

2) Открыть в нём модель мотора

3) Зайти во вкладку Design->Configure Power Rails...

и в выпадающем списке для VCC/VDD выставить напряжение 3.3В

4) Написать программу на языке С согласно стандарта, описанного в файле. Программа должна обеспечивать следующий функционал:

- Приём и передачу данных по UART
- Выдачу напряжения на обмотки двигателя
- Чтение данных с датчика скорости
- Управление скоростью двигателя на основе ПИД-регулятора

Требования к отдельным модулям:

1) Приём и передача данных по UART

Работа с UART должна быть написана в отдельном файле debug.

Каждую секунду текущее значение скорости должно выводиться в терминал.

Если из терминала в микроконтроллер приходит символ “b”, то вывод текущего значения приостанавливается и программа ожидает ввода нового задания

Новое задание вводится через терминал в виде четырехзначного числа со знаком (+0010, -1223) от -2048 до 2047

По окончанию ввода задания или при некорректном вводе программа возвращается в режим выдачи текущего значения.

Если новое задание было корректным, то оно принимается за заданную скорость. Если нет – используется последнее корректное задание. После включения задание 0.

2) Выдача напряжения на обмотки

Работа с выдачей напряжения должна быть написана в отдельном файле motor_voltage.

Должна быть реализована функция void motor_voltage_setVoltage(int16_t), которая позволит выдавать напряжения обоих полярностей на обмотки мотора. Диапазон возможных значений -1000…1000

3) Чтение данных с датчика скорости

Чтение данных с датчика скорости должно быть реализовано в отдельном файле motor_speed.

Должна быть реализована функция int16_t motor_speed_getSpeed(void), которая позволит читать значения скорости от -2048 до 2047.

4) Управление скоростью двигателя на основе ПИД-регулятора

Для управления скоростью должен быть написан ПИД-регулятор в файле control.

ПИД-регулятор должен быть реализован в функции int16_t control_run(int16_t), которая ожидает на входе ошибку управления и возвращает величину управляющего сигнала.

Используемая периферия: ADC1, USART1, TIM1, GPIO