Техническое задание на разработку контроллера для нетривиальных станков с ЧПУ

# Неформальное ТЗ

## Чего я хочу лично

Хочу разработать отладочную платку подороже, чтобы

1. Научиться работать с интерфейсами, микросхемами
2. Получить платку, которую можно использовать для прототипирования ПО и проектов
3. Иметь возможность прототипировать разные кинематические штуки ( станки/принтеры ), не тратя много времени на сборку электроники и не опасаясь испортить дорогое оборудование ( ПК, например ).
4. Иметь возможность адаптировать этот проект под техническую задачу. Как по мне проще сперва сделать всё целиком, а затем убирать лишнее, чем добавлять нужное. В данном случае избыточность – хорошая черта

## Что оно должно делать глобально

Контроллер для нетривиальных станков с ЧПУ. Следовательно он должен уметь управлять станком ( посылать сигналы на его приводы ), знать его состояние ( снимать показания с датчиков ). Общаться с ПК по локальной сети.

## Какие качества я хочу от него

Хочу, чтобы он был бескомпромиссным, лишённым полумер. Надёжность, стабильность.  
Безопасность для людей и имущества людей.   
Хочу, чтобы питался от большого диапазона напряжений. Чтобы управлял моторами на хороших скоростях. Чтобы умел отрабатывать аварийные состояния.

## Какие функции я хочу просто словами

## Его функции подробно

# Формальное ТЗ

### Описание

Настоящее ТЗ описывает требования к разработке аппаратно-программного комплекса (платка), компонента станка с числовым программным управлением (станка) с произвольной кинематикой.

Надо где-то обозначить, что есть платка (собсна платка с STM) ,  
то, что питается от неё (экранчик, энкодеры, концевики),  
 а есть другие части (драйвера моторов, моторы, модбас-устройства, компрессор, шпиндель)  
При этом питание большинства внешних устройств (приводы, модбас-устройства) проходит через блок измерения мощности на платке

Платка должна иметь набор функций, необходимый для непринуждённого прототипирования станка, а именно

* Взаимодействие с пользователем посредством ПК и графического пользовательского интерфейса ( ГУЙ)
* Взаимодействие с пользователем посредством встроенной консоли
* Чтение, проверка и интерпретация G-команд
* Управление приводами в соответствии с G-командой или запросом пользователя
* Сбор информации о станке, например
  + Координаты осей
  + Температура инструмента
  + Давление и поток смазывающе-охлаждающей жидкости (СОЖ)
  + Давление воздуха в ресивере компрессора
* Управление устройствами посредством Modbus
* Индикация
* Определение аварийных ситуаций
* Измерение потребляемой мощности
* Питание от большого диапазона напряжений

### Требования к питанию

* Напряжение питания от 15 до 36 В
* Ток потребления не более 3 А
* Защита платки от включения напряжения обратной полярности до 36 В

### Требования к измерению мощности и защите от превышения тока

* Диапазон измеряемого напряжения от 12 до 36 В
* Погрешность измерения напряжения не более 5%
* Диапазон измеряемого тока от 0 до 20 А
* Погрешность измерения тока не более 5%
* Порог срабатывания защиты 20 А
* Время срабатывания защиты не более 30 мс там электромагнитное реле, может отпустить не сразу

### Требования к скорости перемещения

* Для Декартовой кинематики линейная скорость инструмента не менее 2000 мм/мин
* Для других кинематик этот параметр не определён

Вщ какой-то глупый параметр. Всё очень зависит от механики. Будет компромисс между точностью и скоростью. Надо придумать, как обобщить

### Требования к точности

### Требования к точности

# Глоссарий

* G-код: (англ. G-code) условное именование языка программирования устройств с числовым программным управлением. Называется так, потому что команды имеют вид  
  G01 X0 Y0 Z10 F300. G01 – команда линейной интерполяции, XYZ – конечные координаты, F – скорость, мм/мин