

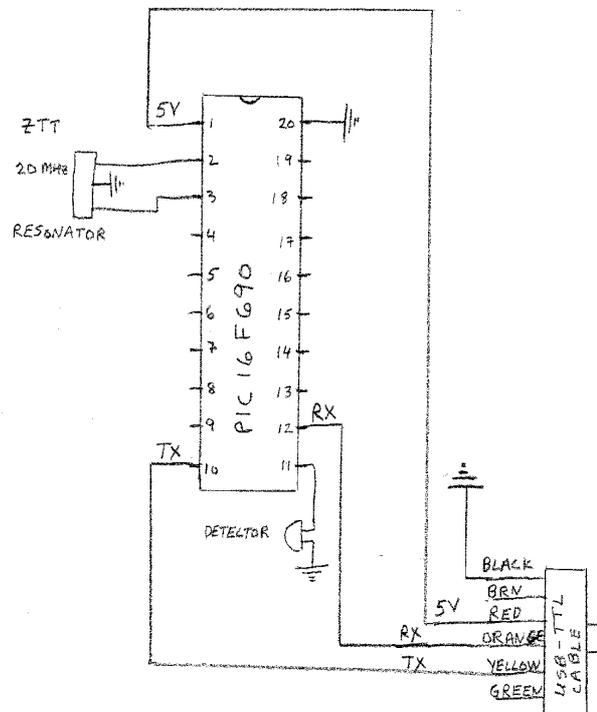
# Настройка лазерного затвора

Страница все еще находится в стадии разработки

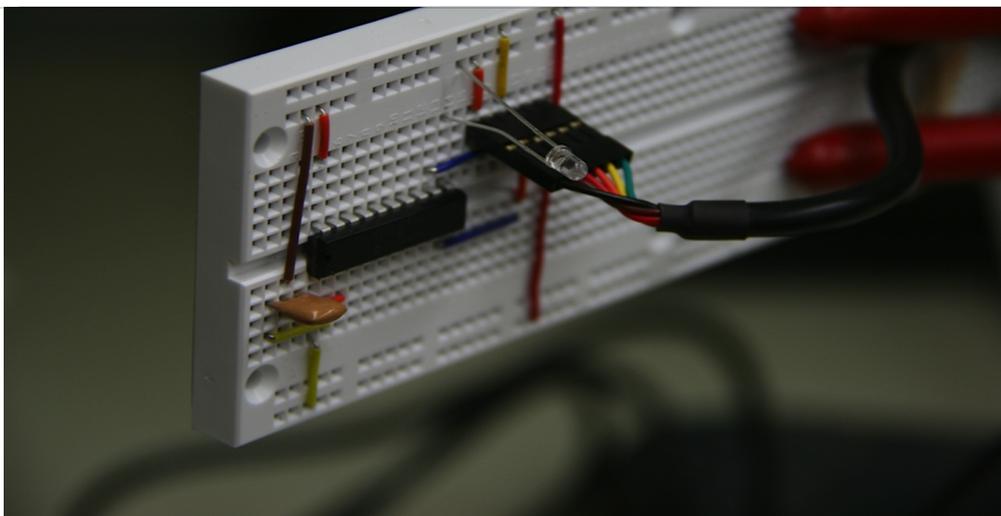
На физическом факультете Калифорнийского университета в Помоне мы разработали недорогой точный лазерный регулятор времени для использования в физических экспериментах. Мы используем микросхему PIC16F690 для синхронизации и передачи данных через USB на ПК. Для запуска программного обеспечения вы можете использовать приведенный ниже файл изображения. Сначала мы объясним, как сконструировать ворота, а затем как запустить программное обеспечение. Общая стоимость системы составляет менее 30 долларов, причем большая часть стоимости приходится на кабель USB-TTL (20 долларов).

## Строительство Ворот

Вентиль можно сконструировать, разместив компоненты на макетной плате без пайки или припаяв компоненты на перфорированной плате. Соединения (с резонатором 20 МГц) показаны на рисунке справа:



Нажмите [здесь](#), чтобы просмотреть список [запчастей](#). Ниже показано изображение макетной платы с подключениями:



Микросхема PIC может быть загружена с [помощью timer3.asm](#) (тактовая частота 4 МГц) или [timer3fast.asm](#) (тактовая частота 20 МГц). При тактовой частоте 4 МГц на PIC16F690 максимально **короткое измеренное время блокировки составляет 2 мс**. Для времени блокировки, превышающего 2 мс, точность времени блокировки составляет 2 микросекунды. Если время блокировки меньше 2 мс, вам нужно будет использовать нашу более быструю версию с резонатором 20 МГц, [timer3fast.asm](#). С резонатором 20 МГц время блокировки может составлять всего 0,4 мс с точностью до 0,4 микросекунды.

## Программное обеспечение

1. **constacc**: Эта программа может быть использована, если ожидается постоянное ускорение, например, падение с лестницы, машина Этвуда, динамика вращения. Введите "z" для калибровки. После ввода "z" вам будет предложено ввести количество блокировок для калибровки и расстояние между первой и последней блокировкой калибровки. Чтобы проверить, работает ли затвор, введите "d", введите и заблокируйте затвор рукой n раз, чтобы проверить, записываются ли данные. Если ворота работают, введите "d", введите, чтобы собрать данные. Исходным кодом для быстрой синхронизации является [constaccf.c](#). Вам нужно запустить код в root.
2. **tandv**: Эта программа собирает данные о времени и скорости. Введите "n", enter, чтобы ввести количество точек данных. Введите "z", enter, чтобы ввести ширину блокировки объекта, который блокирует ворота. Вы можете записать данные в файл в памяти, набрав "w". Файл с именем "tvout.txt" создается в папке "файл", которую можно перенести на карту памяти. Исходным кодом для быстрой синхронизации является [tandvf.c](#). Вам нужно запустить код в root.
3. **маятник**: эта программа записывает время и скорость и выводит данные на экран каждый раз, когда ворота блокируются. Программа предназначена для записи данных с качающегося маятника, поскольку промежутки времени между блокировками относительно велики. Введите "n", enter, чтобы ввести количество точек данных. Введите "z", enter, чтобы ввести ширину блокировки объекта, который блокирует ворота. Как и в случае с tandv, вы можете записать данные в файл в памяти, набрав "w". Файл с именем "pendout.txt" создается в папке "файл", которую можно перенести на карту памяти. Исходный код - [pendulumf.c](#). Вам нужно запустить код в root.
4. **linefitjs.html** это программа линейной подгонки, которая дает наклон и пересечение, а также неопределенность в этих двух параметрах. Щелкните по

Когда вы выключите компьютер, вас спросят, хотите ли вы сохранить свои настройки. Переместите курсор в **положение НЕ СОХРАНЯТЬ**, и на вашем жестком диске ничего не будет сохранено.

[Домашняя страница Сигела](#)

[Физический факультет](#) | [Колледж естественных наук](#)