**Стенд отладки и программирования САМ для массового производства**

* **Что автоматизируем на стенде**
  + - Включить/сбросить питание стенда.
    - Сбросить питание USB-портов, тем самым перезагрузив периферийные устройства на шине USB, подключенные к стенду.
    - Прошить FPGA.
    - Управление VIO.
    - Загрузка программного обеспечения в CPU/STM.
* **Основной стенд:**
  + **ПК или одноплатный компьютер (SBC):**
    - ПК на базе Windows/Linux
    - Raspberry Pi 4 или подобный SBC (для более компактного решения).
  + **Управление портами:**
    - USB хаб на 10 портов (с питанием).
    - USB-JTAG адаптеры (10 штук) - Altera USB Blaster, Terasic USB Blaster, или их аналоги.
* **Интерфейсная плата (необходима разработка):**
  + Специальная плата для коммутации 10 плат CAM модулей с USB-JTAG адаптерами.
  + Защитные цепи от перенапряжения и короткого замыкания.
  + Разъемы для подключения CAM модулей (прищепка с контактами).
  + Светодиодная индикация статуса программирования каждого модуля.
* **Питание:**
  + Блок питания достаточной мощности для всех 10 плат и USB-хаба.

**2. Программное обеспечение (ПО):**

* **На ПК или SBC (управляющее ПО):**
  + **Скрипт на Python/Bash:**
    - Обнаружение подключенных JTAG адаптеров.
    - Запуск программирования на каждом адаптере в параллель.
    - Логирование процесса программирования (статус, время).
    - Возможность удаленного управления.
    - Сохранение отчетов.
  + **API для работы с JTAG программаторами:**
    - Использование библиотек Altera или других производителей.
  + **Скрипт для обновления прошивки:**
    - Автоматическое получение последней прошивки с сервера.
  + **GUI (опционально):**
    - Интерфейс для мониторинга процесса программирования.
    - Возможность ручного управления.
* **На сервере (удаленное управление):**
  + **REST API:**
    - Для управления процессом программирования (запуск, остановка).
    - Для получения статуса, логов, отчетов.
  + **База данных:**
    - Для хранения информации о CAM модулях, их серийных номерах.
  + **Интерфейс пользователя:**
    - Веб-интерфейс для мониторинга и управления.

USB HUB PPPS

JTAG Altera

SNMP

Hwserver (Quartus)

Uhubctl

Py scrypt

Terminal

CPU JTAG

FPGA JTAG

JTAG J-Link

UART-USB

UART

CAM

CAM

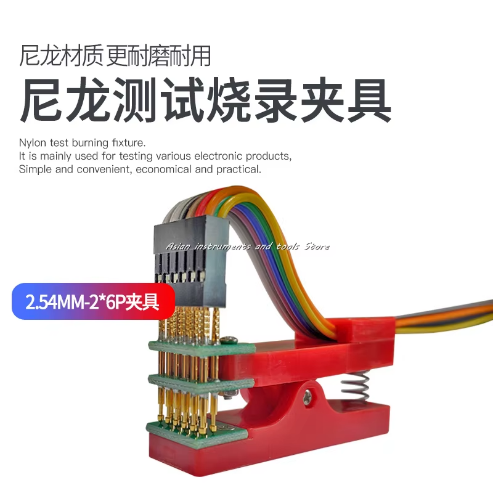
CAM

CAM

CAM

CAM

PDU



Возможный вариант крепления программатора USB Blaster к плате КАМ модуля. На прищепку. Через USB-hub подключаем до 10 плат.



* + Функциональный стенд для программирования
* **Удаленный доступ:**
  + Защищенный VPN или SSH туннель для удаленного доступа к стенду.
  + Многофакторная аутентификация.

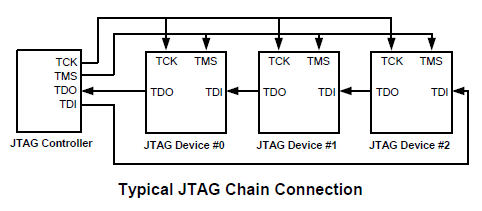
**Важные детали:**

* **Серийные номера:** Присвоение уникальных серийных номеров каждому модулю на этапе производства и использование их для идентификации при программировании.
* **Проверка:** написать тестовую прошивку для проверки работоспособности CAM модулей после прошивки.
* **Безопасность:** Защита прошивок от несанкционированного доступа.
* **Масштабируемость:** продумать возможность увеличения количества программируемых плат на стенде.

**Следующие шаги:**

1. **Детальная разработка:** Составление подробного ТЗ на все компоненты стенда.
2. **Выбор компонентов:** Выбор конкретных моделей контроллеров, USB-JTAG адаптеров, блоков питания и т.д.
3. **Разработка интерфейсной платы:** Создание чертежей и изготовление интерфейсной платы.
4. **Написание ПО:** Разработка скриптов и веб-интерфейса.
5. **Сборка и тестирование:** Сборка стенда и тестирование в реальных условиях.
6. **Оптимизация:** Оптимизация процесса программирования для максимальной производительности.
7. **Масштабирование:** Увеличение количества программируемых плат на стенде (опционально).

**Что по соединению**



**Расчеты времени на программирование модулей (разом на стенде 10 плат)**

* **Объем программирования:** 20,000 - 30,000 модулей в месяц.
* **Время программирования 10 модулей:** 10 минут.
* **Время монтажа/демонтажа 10 модулей:** 3 минуты.
* **Общее время на цикл (10 модулей):** 10 минут + 3 минуты = 13 минут.
* **Рабочий день:** 8 часов.

**Расчеты времени**

1. **Модулей в день (минимально):**
   * 20,000 модулей / 30 дней = 666.67 ≈ 667 модулей в день.
2. **Модулей в день (максимально):**
   * 30,000 модулей / 30 дней = 1000 модулей в день.
3. **Количество циклов программирования в день (минимально):**
   * 667 модулей / 10 модулей на стенд = 66.7 ≈ 67 циклов.
4. **Количество циклов программирования в день (максимально):**
   * 1000 модулей / 10 модулей на стенд = 100 циклов.
5. **Время на цикл (программирование + монтаж/демонтаж):** 13 минут
6. **Время программирования в день (минимально):**
   * 67 циклов \* 13 минут = 871 минут или 14.52 часов.
7. **Время программирования в день (максимально):**
   * 100 циклов \* 13 минут = 1300 минут или 21.67 часов.
8. **Количество циклов в 8-часовой рабочий день:**
   * 8 часов \* 60 минут / 13 минут = 36.92 циклов за 8 часов.
9. **Необходимое количество стендов (минимум):**
   * 67 циклов / 36.92 цикла на стенд = 1.81 стенда, для стабильности округляем до **2** стендов.
10. **Необходимое количество стендов (максимум):**
    * 100 циклов / 36.92 цикла на стенд = 2.71 стенда, для стабильности округляем до **3** стендов.

**Расчет необходимого количества персонала:**

* **Операторы стендов:**
  + Предположим, что один оператор может одновременно обслуживать 2 стенда.
  + Для **2** стендов потребуется **1** оператор.
  + Для **3** стендов потребуется **2** оператора.
* **Смены:**
* Если работа ведется 8 часов в день и 5 дней в неделю, то нужна одна смена. Для **2х** стендов нужно **1** оператор. Для **3х** стендов нужно **2** оператора
* **Инженеры:**
  + Инженер для настройки, обслуживания стендов и решения проблем.
  + На начальном этапе достаточно **1** инженера на все стенды.

**Итого:**

* **Количество стендов:**
  + **2-3** стенда для программирования (минимум 2, для надежности лучше 3).
* **Персонал:**
  + **Операторы:**
    - 1 оператор для 2х стендов (одна смена).
    - 2 оператора для 3х стендов (одна смена).
  + **Инженер:** **1** (на все стенды).
  + **Итого:** **2-3** специалиста (с учетом операторов и инженера)