



Московский институт электроники и  
математики им А.Н. Тихонова

Лаборатория  
САПР

Москва  
2024

# Удаленный стенд на основе микроконтроллера МК32 Амур

Презентацию подготовил студент МСМТ243:  
Дьяконов Роман Юрьевич



# MIK32 Амур – первый Отечественный RISC-V микроконтроллер

Предназначенный для  
встраиваемых систем и интернета  
вещей, MIK32 Амур полностью  
произведен в России на  
предприятии Mikron





## Цель и поставленные задачи

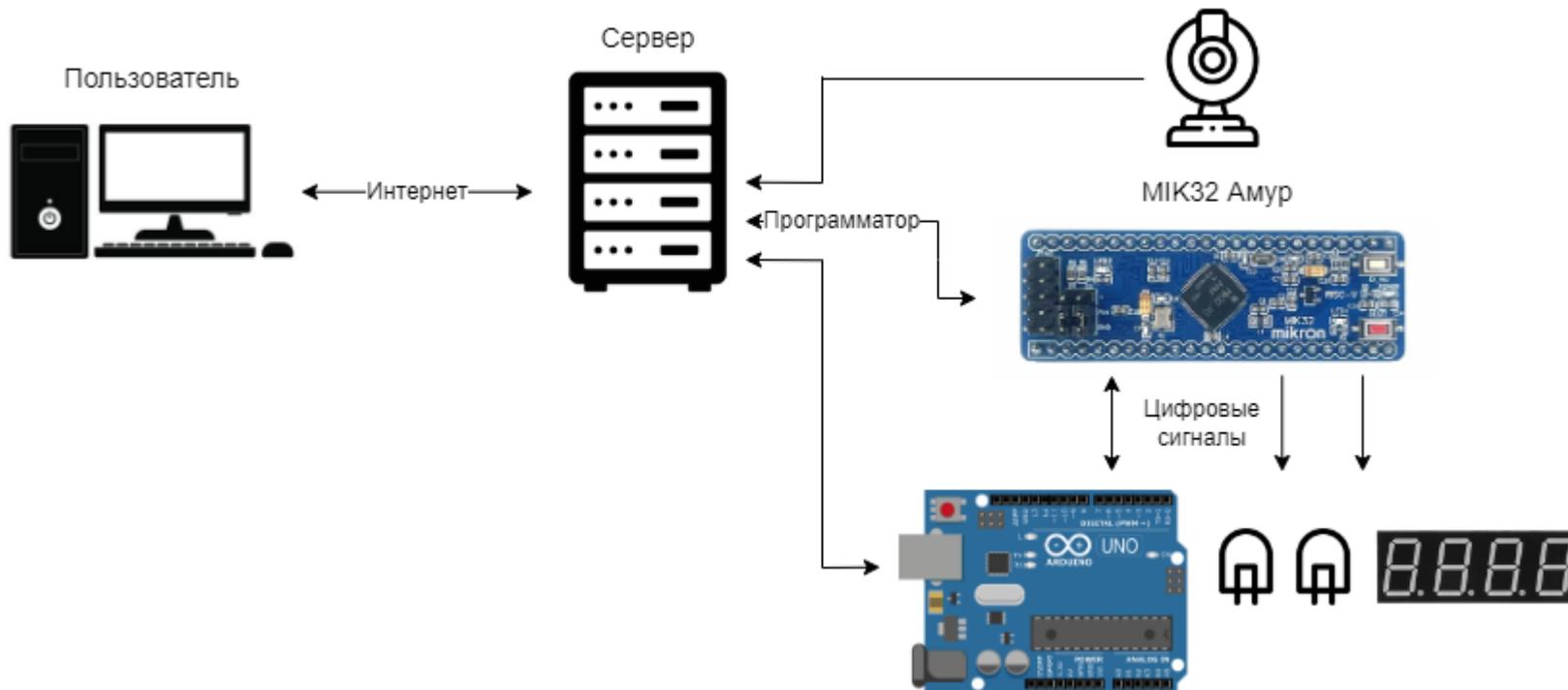
### Цель работы:

- Разработка учебного стенда на основе MIK32 Амур с возможностью удаленной работы.

### Задачи:

- Спроектировать стенд с микроконтроллером, периферией, камерой и устройством для обмена сигналами,
- Разработать программное обеспечение для работы сервера и его взаимодействия со стендом,
- Наладить работу с отладочной платой MIK32 Амур и создать инструкцию для работы с ней.

## Удаленный стенд на основе MIK32 Амур





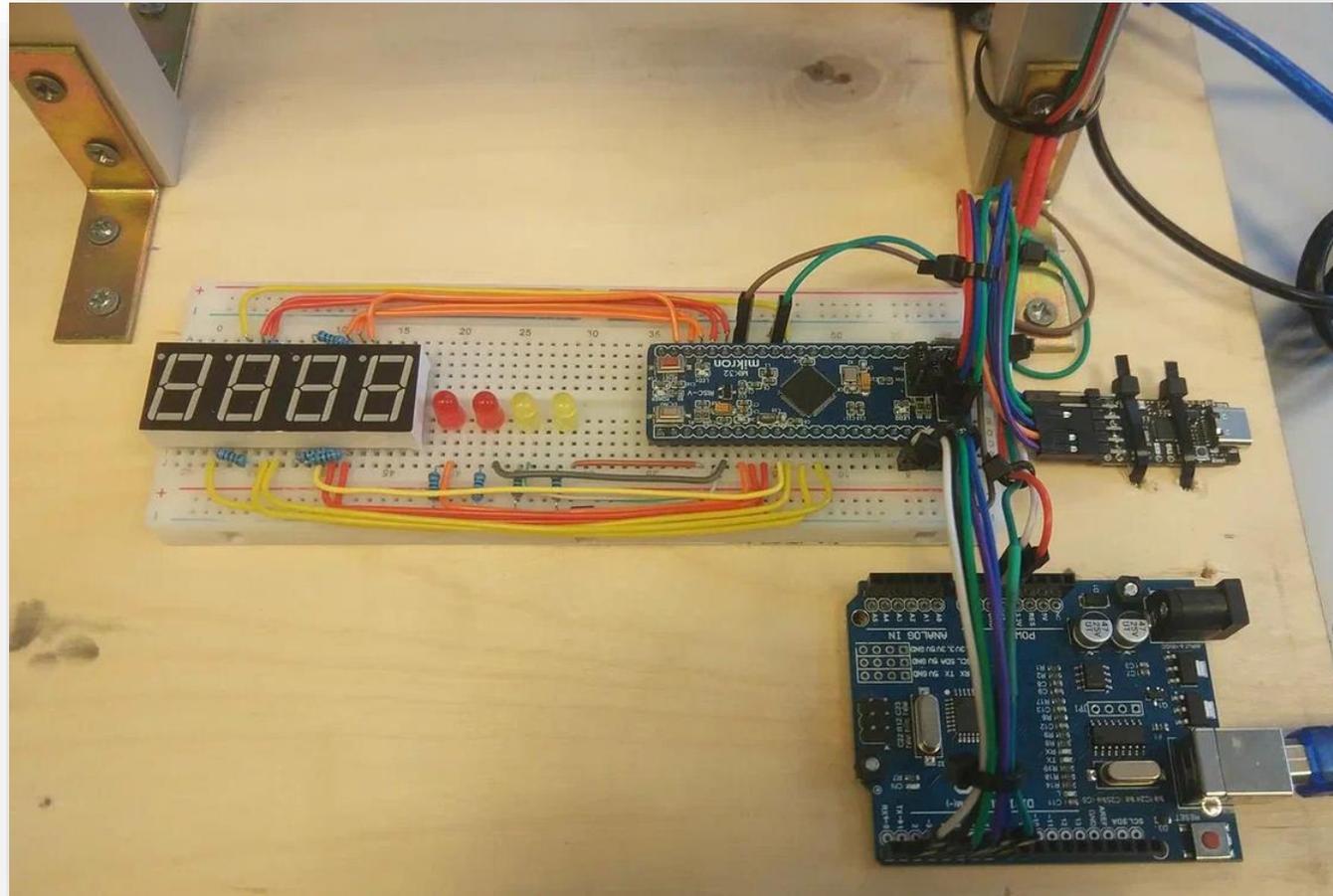
## Удаленные стенды САПР

- Удаленный доступ к ПЛИС учебной лаборатории САПР в МИЭМ





## Внешний вид устройства

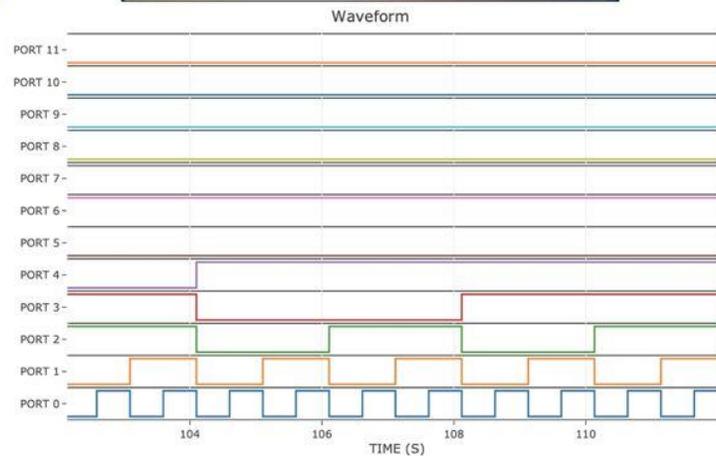
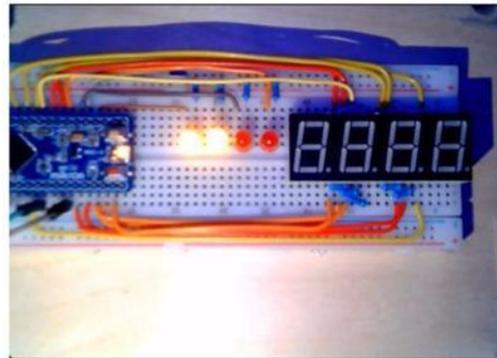


# Веб-интерфейс удаленного стенда

## Удалённый стенд MIK32 “Амур”

Виртуальные входы/выходы:

Входы	Выходы
PORT 0 <input type="checkbox"/>	PORT 6 <input checked="" type="checkbox"/>
PORT 1 <input type="checkbox"/>	PORT 7 <input checked="" type="checkbox"/>
PORT 2 <input type="checkbox"/>	PORT 8 <input checked="" type="checkbox"/>
PORT 3 <input type="checkbox"/>	PORT 9 <input checked="" type="checkbox"/>
PORT 4 <input type="checkbox"/>	PORT 10 <input checked="" type="checkbox"/>
PORT 5 <input checked="" type="checkbox"/>	PORT 11 <input checked="" type="checkbox"/>



### Загрузка программы

```
PCC_OscInit.KernelDriver = 0;
// Поправочные коэффициенты
PCC_OscInit.HSI32KCalibrationValue = 128;
PCC_OscInit.LSI32KCalibrationValue = 128;

// Инициализация
HAL_PCC_Config(&PCC_OscInit);
}

void GPIO_Init() { // Функция инициализации GPIO
// Значение по умолчанию
GPIO_InitTypeDef GPIO_InitStructure = {0};

// Включение тактирования на GPIO
__HAL_PCC_GPIO_CLK_ENABLE();
__HAL_PCC_GPIO1_CLK_ENABLE();
__HAL_PCC_GPIO2_CLK_ENABLE();
__HAL_PCC_GPIO_IRQ_CLK_ENABLE();

// Выбор настроек входной группы GPIO 0
GPIO_InitStructure.Mode = HAL_GPIO_MODE_GPIO_OUTPUT;
GPIO_InitStructure.Pull = HAL_GPIO_PULL_NONE;
GPIO_InitStructure.Pin = GPIO_PIN_0 | GPIO_PIN_1 | GPIO_PIN_2 | GPIO_PIN_3 |
GPIO_PIN_4 | GPIO_PIN_7;
HAL_GPIO_Init(GPIO_0, &GPIO_InitStructure);

// Выбор настроек выходной группы GPIO 1
GPIO_InitStructure.Mode = HAL_GPIO_MODE_GPIO_OUTPUT;
GPIO_InitStructure.Pull = HAL_GPIO_PULL_NONE;
GPIO_InitStructure.Pin = GPIO_PIN_6 | GPIO_PIN_7 | GPIO_PIN_8 | GPIO_PIN_9 |
GPIO_PIN_10 | GPIO_PIN_11 | GPIO_PIN_12 | GPIO_PIN_13 |
GPIO_PIN_14 | GPIO_PIN_15;
HAL_GPIO_Init(GPIO_1, &GPIO_InitStructure);

// Выбор настроек выходной группы GPIO 2
GPIO_InitStructure.Mode = HAL_GPIO_MODE_GPIO_INPUT;
GPIO_InitStructure.Pull = HAL_GPIO_PULL_NONE;
GPIO_InitStructure.Pin = GPIO_PIN_0 | GPIO_PIN_1 | GPIO_PIN_2 | GPIO_PIN_3 |
GPIO_PIN_4 | GPIO_PIN_5 | GPIO_PIN_8 | GPIO_PIN_7 |
GPIO_PIN_8 | GPIO_PIN_9 | GPIO_PIN_10 | GPIO_PIN_11 |
GPIO_PIN_12 | GPIO_PIN_13 | GPIO_PIN_14 | GPIO_PIN_15;
HAL_GPIO_Init(GPIO_2, &GPIO_InitStructure);
}
```

Загрузить DONE

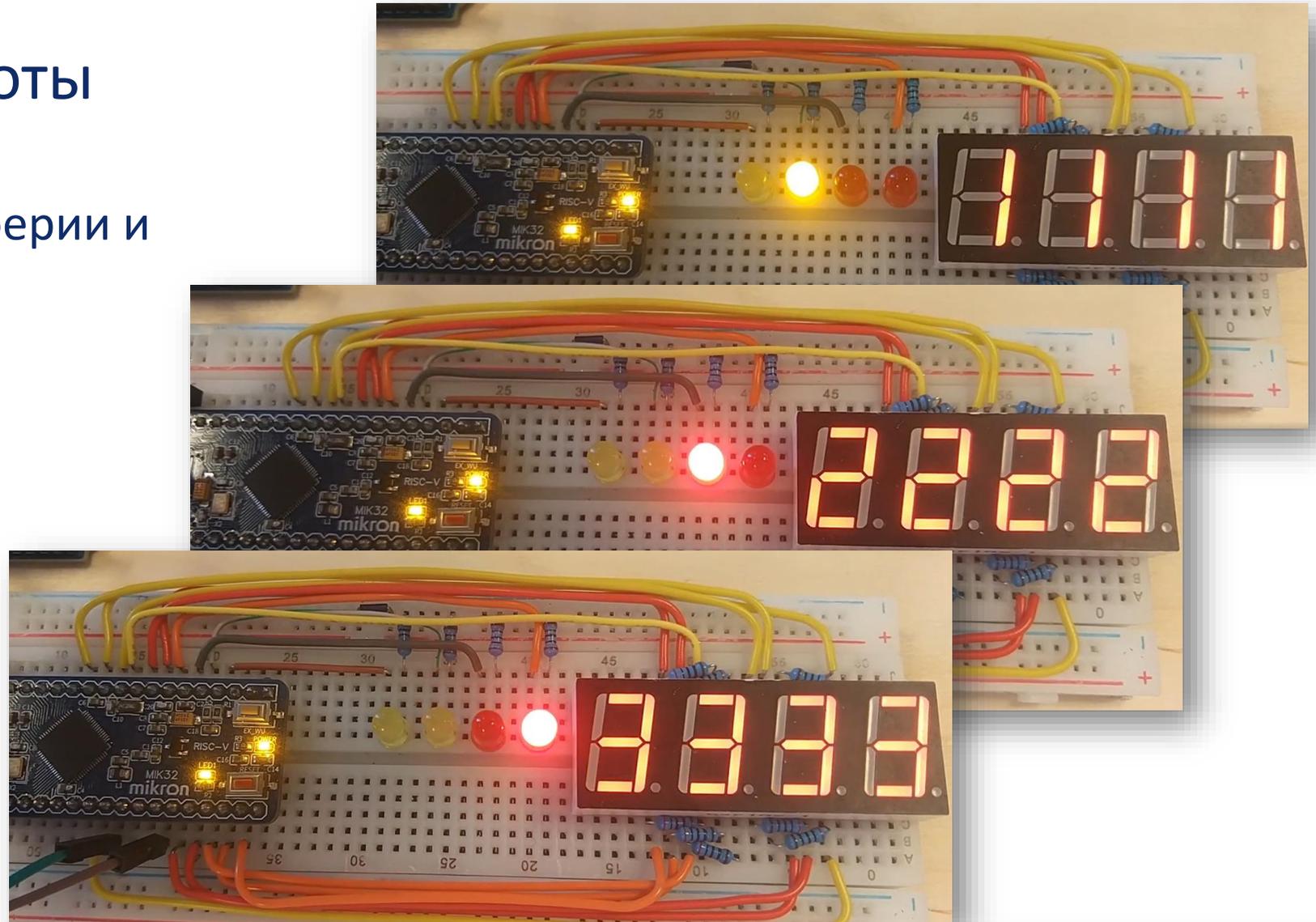
### Лог загрузки

```
Processing mik32v0 [platform: MIK32; board: mik32v0; framework: framework-mik32v0-sdk; board
-----
ldscript: /Users/ax-sayfulin/.platformio/packages/framework-mik32v0-sdk/shared/ldscripts/ram.ld
CONFIGURATION: https://docs.platformio.org/page/boards/mik32/mik32v0.html
PLATFORM: Mikron MIK32 Platform (0.2.1) > MIK32V0 Generic Board
HARDWARE: MIK32V0 32MHz, 16KB RAM, 8KB Flash
DEBUG: Current (ftdi) On-board (ftdi)
PACKAGES:
- framework-mik32v0-sdk @ 0.2.0
- tool-mik32-uploader @ 0.1.0
- tool-openocd @ 3.1200.0 (12.0)
- toolchain-riscv @ 1.80300.190927 (8.3.0)
ldscript: /Users/ax-sayfulin/.platformio/packages/framework-mik32v0-sdk/shared/ldscripts/ram.ld
LDF: Library Dependency Finder -> https://bit.ly/configure-pio-ldf
LDF Modes: Finder ~ chain, Compatibility ~ soft
```



## Демонстрация работы

Для показа работы периферии и  
ознакомления с  
программированием  
микроконтроллера  
в репозитории находится  
инструкция и  
демо-программа





## Дальнейшее развитие

После защиты ВКР планировалось:

- добавление дополнительной периферии (сервопривод, ЖК-дисплей и др.);
- проектирование корпуса в едином для МИЭМ стиле для встраивания в серверную стойку;
- авторизация через почту МИЭМ/ВШЭ через «единое окно»;
- режим асинхронного выполнения;
- разработка платы стенда и ее заказ.





# Спасибо за внимание!

