Программируемый логический контроллер KRAX PLC-932

Инструкция пользователя

Оглавление

Обзор ПЛК KRAX PLC-932

ПЛК KRAX 932 на базе MicroPython - это контроллер, который использует MicroPython в качестве языка программирования.

MicroPython - это порт Python 3 для микроконтроллеров. Он представляет собой интерпретатор Python, который работает на аппаратном уровне. **Аппаратные свойства ПЛК**

•**Процессор:** двухъядерный процессор Xtensa LX6 с тактовой частотой 240 МГц.

•**Оперативная память:** 32 МБ ОЗУ и 8 МБ PSRAM.

•**Внутренняя память:** 4 МБ флэш-памяти.

•**Энергонезависимая память EEPROM:** 8 Кбайт для сохранения переменных

•**Интерфейсы:** Wi-Fi, Bluetooth, Ethernet, UART.

Модульная архитектура

ПЛК KRAX 932 имеет модульную архитектуру, что позволяет расширять его функциональность. Для подключения сигналов исполнительных устройств доступны следующие виды модулей расширения:

•**Модуль дискретного входа DO-430:** позволяет подключать 8 дискретных входных сигналов. Каналы поддерживают режим подсчета импульсов в режиме 8 битного счетчика. Используемый объем памяти ввода/вывода до 9 байт

•**Модуль дискретного выхода DO-530:** позволяет подключать 8 дискретных выходных сигналов. Используемый объем памяти ввода/вывода 1 байт

•**Модуль аналогового входа AI-455:** позволяет подключать 4 аналоговых входных сигнала 4-20 мА. Используемый объем памяти ввода/вывода 16 байт

Взаимодействие между ПЛК и модулями ввода-вывода происходит по беспроводному интерфейсу ESP-NOW. Скорость обмена настраивается. Заводские настройки приведены в таблице:

Скорость обмена	56 МБит/сек
Отклик модуля расширения	5 мсек.
Максимальный объем памяти ввода/вывода	200 байт

Дополнительная информация

•**PSRAM (Pseudo Static RAM)** - это тип оперативной памяти, которая имитирует статическую память. Она имеет более высокую скорость доступа, чем динамическая память, что делает её более подходящей для задач, требующих высокой производительности.

•**512 КБ ОЗУ и 8 МБ PSRAM** обеспечивают достаточную память для хранения кода, данных и буферов.

•**Wi-Fi, Bluetooth, Ethernet** обеспечивают широкий спектр возможностей подключения.

Преимущества ПЛК на базе MicroPython

- **Простота программирования:** MicroPython - это простой язык программирования, который легко освоить. Это делает его привлекательным для новичков в области промышленной автоматизации.
- **Открытость:** MicroPython - это открытый исходный код, что означает, что его можно свободно использовать и изменять. Это позволяет пользователям адаптировать ПЛК для своих конкретных потребностей.
- **Гибкость:** MicroPython - это мощный язык программирования, который позволяет реализовать сложные алгоритмы управления.

Области применения

ПЛК на базе MicroPython могут использоваться в различных областях, включая:

- **Промышленная автоматизация:** ПЛК можно использовать для управления производственным оборудованием, такими как станки и конвейеры.
- **Интернет вещей:** ПЛК можно использовать для управления устройствами Интернета вещей, такими как датчики и исполнительные устройства.
- **Умные дома:** ПЛК можно использовать для автоматизации домов и квартир.

Заключение

ПЛК на базе MicroPython - это привлекательный вариант для тех, кто ищет простой, открытый и гибкий контроллер. Они могут использоваться в различных областях, включая промышленную автоматизацию, Интернет вещей и умные дома.

Установка и настройка ПЛК

Технические требования

Электропитание

- Питание: 24 В постоянного тока
- Диапазон входного напряжения: 12-32 В постоянного тока
- Потребляемая мощность: 10 Вт

Интерфейсы

- Ethernet: 1 порт
- USB: 1 порт
- WiFi
- Bluetooth, BLE

Ввод/вывод

В зависимости от вашей задачи вам потребуются модули серии KRAX для подключения сигналов от датчиков и исполнительных механизмов.

Другие требования

Температура окружающей среды: -20 до 55 °C

Влажность окружающей среды: 5 до 95 %

Подробное описание технических требований

Электропитание

ПЛК питается от источника постоянного тока напряжением 24 В. Диапазон входного напряжения составляет 20-28 В постоянного тока. Потребляемая мощность ПЛК составляет 10 Вт.

Интерфейсы

ПЛК имеет следующие интерфейсы:

USB для подключения к компьютеру для программирования и взаимодействия с MicroPython REPL

Ethernet для подключения к сети, для взаимодействия со средой визуализации (SCADA), программирования, взаимодействия с MicroPython REPL

WiFi — доступна для использования программой на ПЛК, для программирования, доступа к MicroPython REPL.

Bluetooth — для использования программой на ПЛК (на усмотрение разработчика, например для управления термо-принтером).

Ввод/вывод

ПЛК имеет 3 дисретных входа в виде 2-х джамперов и одной кнопки.

Сигналы от датчиков подключаются к модулям расширения, обмен с которыми происходит по беспроводному интерфейсу ESP-NOW.

Установка

Для быстрого старта потребуется провод MicroUSB, при помощи которого ПЛК подключается к USB порту ПК.

Для разработки программного обеспечения ПЛК рекомендуется компьютер с операционной системой Windows,Linux или MacOS со средой разработки приложений на языке Python и утилита для обмена upydev.

Для разработки приложений на языке Python можно использовать различные среды разработки, такие как Eclipse, PyCharm, Visual Studio Code и другие. Мы будем использовать VS Code с дополнительным расширением, облегчающим обмен файлами с ПЛК.

Также необходимо установить Python3.

VS Code

Visual Studio Code - это бесплатная и open-source среда разработки, поддерживающая широкий спектр языков программирования, включая Python. VS Code предлагает множество функций, которые упрощают и ускоряют процесс разработки, таких как IntelliSense, отладка и интеграция с Git.

Установка VS Code

Для установки VS Code необходимо перейти на официальный сайт VS Code (Microsoft) и загрузить установочный файл. После загрузки установочного файла необходимо запустить его и следовать инструкциям на экране.

Настройка VS Code для Python и ПЛК

После установки VS Code необходимо настроить его для работы с языком Python. Для этого необходимо выполнить следующие действия:

- Откройте VS Code.
- В меню Extensions (Расширения) введите "python".
- Установите расширение Python от Microsoft.
- В меню Extensions выберите Install from VSIX и выберите файл krax-io-X.X.X.vsix, где X.X.X — версия. Скачать его можно тут
- В меню File (Файл) выберите пункт Preferences (Параметры).
- В разделе Languages & Frameworks (Языки и фреймворки) выберите Python.
- В разделе Python: Default interpreter (Python: интерпретатор по умолчанию) укажите путь к интерпретатору Python.

Установка утилиты upydev для обмена файлами с ПЛК

Для установки пакета upydev через pip необходимо выполнить следующие действия:

Убедитесь, что у вас установлена последняя версия pip. Для этого выполните следующую команду:

Откройте терминал или командную строку.

```
pip install --upgrade pip
```

Введите следующую команду для установки пакета upydev:

```
pip install upydev
```

Подождите, пока pip загрузит и установит пакет upydev.

После установки пакета upydev вы сможете использовать для взаимодействия с ПЛК. Подробнее см на сайте разработчиков upydev https://upydev.readthedocs.io/en/latest/

Проверка установки

После установки пакета upydev вы можете проверить его, выполнив следующую команду:

```
pip show upydev
```

Эта команда выведет информацию о пакете upydev, включая его версию.

Настройка

Если все выше прошло успешно, то в коммандной строке или терминале будет доступна утилита upydev.

```
upydev
Должно выдать:
upydev: no device configured (upydev_.config file not found)
usage:  config [-h] [-t T] [-p P] [-g] [-gg] [-@ @] [-zt ZT] [-sec]
```

Подключение к ПЛК с помощью USB

Этот метод удобен тем что ПЛК может работать от USB без внешнего блока питания 24В и тем что его нельзя отключить из ПЛК (Тогда как Ethernet, WiFi, Bluetooth можно отключить из программы ПЛК или просто забыть его IP адрес)

Подключим ПЛК к USB порту при помощи кабеля MicroUSB. В системе появится USB-COM порт (в Windows он называется COMX, в Linux это /dev/ttyUSBX где X — номер порта, в MacOS /dev/tty.usbserial). Пусть в

диспетчере устройств вы определили, что X = 1. Тогда выполним настройку подключения к ПЛК:

```
upydev config -t COM1 -p 115200
```

Теперь можно например посмотреть файлы, которые находятся в ПЛК

```
upydev ls
Должно выдать что-то вида:
boot.py          io.csv          krax.dat          krax.json
main.py          persist.json    task.py           webrepl_cfg.py
```

Теперь вы можете используя комманды upydev получать/отправлять файлы в ПЛК.

Подключение к ПЛК с помощью Ethernet

ПЛК по умолчанию имеет IP адрес 192.168.2.10 (Ethernet), 192.168.4.1 (WiFi). Их можно изменить в файле настроек krax.json. Пароль для подключения 115200

```
upydev config -t 192.168.2.10 -p 115200
```

Подключение к ПЛК модулей ввода-вывода

Обмен с модулями ввода/вывода производится по беспроводному интерфейсу ESP-NOW, но для начала необходимо ПЛК «познакомить» с его модулями (для разработки программы в дальнейшем важен порядок модулей, в котором они будут опрашиваться). Для этого необходимо:

1. Запустить в ПЛК предустановленную программу kx_bonjour. Сделать можно это при помощи upydev (или из среды VS Code, см ниже)

```
upydev run kx_bonjour
Running kx_bonjour...
Staring node configuration with id=1. init= {'rate': 12, 'iface': 0, 'flags': 0, 'hostname':
'krax'}
```

2. Нажать и удерживать кнопку WPS пока синий светодиод WPS не замигает.

3. На модулях ввода вывода если синий светодиод WPS горит нажать 2 раза кнопку WPS или один раз если не горит. Светодиод WPS начнет быстро мигать (перешли в режим «знакомства»).

4. Нажать WPS на ПЛК («представиться» модулям). Светодиод WPS на модулях начнет мигать с низкой частотой («улышал» ПЛК). Если какой то модуль не «услышал» или не находился в режиме знакомства то можно повторить пока все модули не будут медленно мигать синим светодиодом.

5. По порядку, в котором необходимо опрашивать модули ввода-вывода, нажимаем кнопку WPS на модулях. При этом светодиод WPS начнет снова быстро мигать, а в момент нажатия на ПЛК проморгнет светодиод COMM (ПЛК «увидел» новый модуль).

6. Завершаем процедуру знакомства длительным нажатием WPS на ПЛК (более 1 сек)

Программирование ПЛК

Разработка программы для ПЛК не отличается от разработки проекта на языке Python. Для удобства в среде VSCode устанавливается расширение krax-io-X.X.X.vsix, при помощи которого создается структура каталогов и скелет проекта, производится обмен с ПЛК (обмен файлами проекта).

Минимальный проект состоит из:

task.py - точка входа в программу ПЛК
io.csv - расположение по каналам модулей ввода/вывода переменных
krax.json - конфигурация модулей и служебные параметры

Быстрый старт

HelloWorld на ПЛК

Работа с ПЛК происходит через комманды расширения KRAX.IO (которые преобразуются в вызов утилиты upydev с разными аргументами). Вызов списка доступных комманд можно щелчком левой кнопки мыши в строке состояния VSCode на «KRAX.IO:», которая должна появится если в каталоге проекта есть файл «.kraxio».

В VSCode вызвать список команд (Ctrl+Shift+P), выбрать KRAX.IO: Getting Started -> New Project -> Указать размещение нового проекта -> Ввести имя проекта HelloWorld. По указанному размещению будет создан каталог проекта со скелетом проекта следующей структуры

src/ - все содержимое этого каталога загружается в PLC

boot.py - запускается при включении PLC

task.py - точка входа в программу PLC

io.csv - расположение по каналам модулей ввода/вывода переменных

krax.json - конфигурация модулей и служебные параметры

.kraxio - служебный файл

boot.py и krax.json удалим, они нам не потребуются.

Файл task.py уже содержит

```
# This is project entry point.
from kx.config import *
print('Starting up KRAX.IO project!')
plc,hw = kx_init( )
instances = [] #here should be listed user defined programs
while True:
    with plc:
        for i in instances:
            i()
```

Комманда KRAX.IO: Target Device. Выбираем COM порт (PLC при подключении для системы видна как последовательный порт (COMX/ttyUSBX/tty.usbserial. Возможно потребуется установить драйвер CP210x)

Комманда KRAX.IO: Upload project. Изменения проекта (если они есть) автоматически загружаются в ПЛК

Комманда KRAX.IO: Run PLC. Текущий файл начнет выполнятся. В терминале появится

```
cd src ; upydev kbi ; upydev reload task; upydev run task
Running task...
Starting up KRAX.IO project!
....
```

Заменим в 3й строке «Starting up KRAX.IO project!» на «Hello World» и повторим Upload project и Run PLC. Готово!