



АВТОМАТТАНДЫРУ ЖӘНЕ БАСҚАРУ
АВТОМАТИЗАЦИЯ И УПРАВЛЕНИЕ
AUTOMATION AND CONTROL

DOI 10.51885/1561-4212_2023_4_228
МРНТИ 55.30.31; 50.41.29

А.Т. Кадыролдина¹, М.Е. Токмин², А.Л. Красавин³, Г.М. Назенова⁴, Ә.Т. Құсайын-Мұрат⁵
Восточно-Казахстанский технический университет им. Д. Серикбаева,

г. Усть-Каменогорск, Казахстан

¹E-mail: akadyroldina@gmail.com*

²E-mail: tokmin@silumin.kz

³E-mail: akrassvin@edu.ektu.kz

⁴E-mail: g_nazenova@mail.ru

⁵E-mail: akussaynmurat@edu.ektu.kz

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ УПРАВЛЕНИЯ РОБОТОМ-МАНИПУЛЯТОРОМ НА БАЗЕ МИКРОКОНТРОЛЛЕРА STM32F407

STM32F407 МИКРОКОНТРОЛЛЕРІНЕ НЕГІЗДЕЛГЕН РОБОТ-МАНИПУЛЯТОРДЫ БАСҚАРУҒА АРНАЛҒАН БАҒДАРЛАМАЛЫҚ ЖАСАҚТАМА ЖАСАУ

DEVELOPMENT OF SOFTWARE TO CONTROL A ROBOT-MANIPULATOR BASED ON STM32F407 MICROCONTROLLER

Аннотация. Промышленные роботы широко используются в производственных подразделениях для выполнения различных задач. Большинство производителей роботов предоставляют программное обеспечение для автономного и онлайн-программирования роботов, но обычно оно продается отдельно. Немногие производители роботов не имеют специального программного обеспечения для моделирования своих роботов. Разработка программного обеспечения для управления роботизированными манипуляторами это значительно сложная задача, поскольку она требует опыта во многих областях, таких как: робототехника, программирование в реальном времени, интеграция оборудования, параллелизм и мн. др. В связи с чем сложно разработать общую программную платформу, поддерживающую разнообразие областей робототехнических исследований и роботизированного оборудования. Несмотря на то, что было создано большое количество роботизированных языков, библиотек и инструментов, они редко используются повторно. То есть многие исследовательские группы разрабатывают свою собственную программную платформу с нуля, потому что существующие платформы слишком негибки в отношении модификаций и слишком сложны для понимания. В данной статье представлена разработка программного обеспечения для управления роботом манипулятором, которая включает в себя программирование в программе CUBE IDE, программирование микроконтроллера, программирование в Ciro's Programming и обработку данных. Показаны достоинства нового программного обеспечения, так как используемое оборудование, предназначенное для его разработки, имеет в распоряжении у исследовательской группы авторов, которое характеризуется простотой в разработке и дешевизной.

Ключевые слова: Робот-манипулятор, робототехника, программирование, микроконтроллер

Аңдатпа. Өндірістік роботтар әртүрлі тапсырмаларды орындау үшін өндіріс бөлімдерінде кеңінен қолданылады. Көптеген робот өндірушілер роботты офлайн және онлайн бағдарламалау бағдарламалық құралын қамтамасыз етеді, бірақ олар әдетте бөлек сатылады. Аздаған робот өндірушілердің роботтарын модельдеуге арналған арнайы бағдарламалық жасақтамасы жоқ. Робототехника, нақты уақыттағы бағдарламалау, аппараттық құралдарды біріктіру, параллельдік және т.б. сияқты көптеген салаларда тәжірибені қажет ететін роботтық манипуляторды басқаруға арналған бағдарламалық жасақтаманы әзірлеу маңызды мәселе болып табылады. Осыған байланысты роботтық зерттеулер мен роботтық

жабдықтардың әртүрлі салаларын қолдайтын ортақ бағдарламалық платформаны жасау қиын. Көптеген роботтық тілдер, кітапханалар мен құралдар жасалғанымен, олар сирек қайталанады. Яғни, көптеген зерттеу топтары өздерінің бағдарламалық платформасын басынан бастап жасайды, себебі бар платформалар өзгертуге тым икемсіз және түсіну өте қиын. Бұл мақалада CUBE IDE бағдарламасында бағдарламалауды, микроконтроллерді бағдарламалауды, Ciroc Programming бағдарламасында бағдарламалауды және деректерді өңдеуді қамтитын роботтық манипуляторды басқаруға арналған бағдарламалық жасақтаманы әзірлеу ұсынылады. Жаңа бағдарламалық қамтамасыз етудің артықшылықтары көрсетілген, сондай-ақ оны әзірлеуге арналған авторлардың зерттеу тобына қол жетімді, пайдаланылған жабдықтар, құрастырудың қарапайымдылығымен және төмен құнымен сипатталады.

Түйін сөздер: Робот манипулятор, робототехника, бағдарламалау, микроконтроллер

Abstract. Industrial robotic arms are widely used in manufacturing departments to perform various tasks. Most robot manufacturers provide offline and online robot programming software, but these are usually sold separately. Few robot manufacturers do not have dedicated software for programming their robots. Developing software to control robotic arms is a very challenging task as it requires experience in many areas such as: robotics, real-time programming, hardware integration, parallelism, etc. It is difficult to develop a common software platform that supports all the variety of areas of robotics research and robotic equipment. While a large number of robotic programming languages, libraries, and tools have been created, they are rarely reused. Many research groups develop their own software platform from scratch because existing platforms are too inflexible to modify and too difficult to understand. This article presents the development of a new software for controlling a robotic arm, which includes programming in the CUBE IDE program, microcontroller programming, programming in Ciroc Programming and data processing. The advantages of the new software are shown, since the equipment used for its development is available to the research group of authors, which is characterized by ease of development and low cost.

Keywords: robotic arm, robotics, programming, microcontroller.

Введение. В настоящее время роботы в основном классифицируются как мобильные и стационарные роботы. Мобильные роботы имеют базу, которая перемещается по земле, воде или воздуху, тогда как у роботов с фиксированной базой одна из ее частей закреплена на земле. Роботы с фиксированной базой далее классифицируются как последовательные системы, параллельные системы, а также как системы древовидного типа, которые представляют собой комбинацию первых двух. Параллельные роботы обычно имеют подвижную платформу, на которой может быть инструмент или объект. Хотя параллельные роботы имеют лучшую жесткость и точность, они имеют меньшие рабочие пространства и подвержены блокировке звеньев робота. С другой стороны, у серийных роботов больше рабочее пространство, но они менее точны. В промышленности серийные роботы используются более широко и называются промышленными манипуляторами [1].

Промышленные роботы или манипуляторы обычно используются для выполнения таких задач, как сборка, шлифовка, захват и размещение, покраска, сварка и т. д. В статье [2] авторами отмечается, что в будущем роботы будут выполнять задачи в декартовом пространстве, где они управляются в своем совместном пространстве. Следовательно, для достижения желаемого движения робота необходимо знать математическую модель, связывающую движение сустава и движение рабочего органа, последнего звена робота. Например, в статье [3] предлагается методология разработки кинематической модели и программного обеспечения для моделирования физического робота. Во-первых, модель с системой автоматизированного проектирования (англ. Computer Added Design, CAD) разрабатывается в программе Autodesk Inventor (Autodesk, США) с последующим автоматическим извлечением ее параметров Денавита-Хартенберга (DH). Пример роботизированной руки Nex Dexter (Nex Robotics Pvt Ltd, Индия) используется для демонстрации предлагаемой методологии. С использованием файлов CAD каждой детали, извлеченных параметров DH и формулировки кинематики, авторами [3] было

разработано и опубликовано программное обеспечение (ПО) для моделирования. ПО имеет возможность выполнять совместные и декартовые толчковые движения в среде моделирования, а также на реальном роботе. Также самая процедура может быть использована для разработки ПО для онлайн- и офлайн-моделирования для любого физического тела.

Роботы используются в средах со статическими и динамическими препятствиями. Кроме того, если движение робота проверяется в ПО для моделирования до того, как оно будет запрограммировано на реальном роботе, усилия и время будут значительно сокращены. Кроме того, симуляция позволяет избежать любого возможного повреждения физического робота. Следовательно, ПО для моделирования робота имеет большое значение [4]. ПО для моделирования роботов в целом подразделяется на автономное и онлайн-моделирование. В первом случае пользователю показывается САД-модель робота и его рабочей ячейки. Любое движение робота в форме шарнирного и декартового толчкового движения может быть предоставлено в качестве входных данных, а 3D-модель САД движется манипулятором робота. Как только желаемое движение достигнуто, движение робота или программа могут быть отправлены на фактический контроллер робота [5]. Во время имитации реальный робот не задействуется и, следовательно, производительность при моделировании не изменяется. Тем не менее, любые динамические препятствия в рабочей ячейке не могут быть идентифицированы и, соответственно, могут изменить движение робота, что делается в режиме онлайн-моделирования. Следовательно, для сред с динамическими объектами, такими как сотрудник-человек, или для нескольких роботов, использующих общее рабочее пространство, необходимо выполнять онлайн-моделирование [6].

Существует несколько программ для моделирования роботов, которые позволяют моделировать роботов в автономном и онлайн-режиме. Некоторые из коммерческих программ включают ABB RobotStudio (ABB Robotics, Швеция, Швейцария), KUKA Sim Pro (KUKA, Германия), библиотеку RoboDK (RoboDK Inc., Канада, Испания), Visual Components (Visual Components, Финляндия) и т.д. Существует несколько бесплатных альтернатив или альтернатив с открытым исходным кодом, таких как ROBOMOSP (США) [1], V-REP (Coppelia Robotics AG, Швейцария) и т. д. Другое такое программное обеспечение - RoboAnalyzer (Индия) [7] - это бесплатное ПО для обучения робототехнике на основе 3D-моделей, разработанное совместно с третьим автором. Это ПО имеет несколько модулей, связанных с прямой кинематикой, обратной кинематикой, обратной и прямой динамикой, а также декартовым планированием движения серийных роботов. Его модуль виртуального робота (VRM) содержит САД-модели более 20 промышленных роботов и может выполнять на них шарнирное и декартово движение. Архитектура VRM сохранена таким образом, что новые модели роботов могут быть легко добавлены и легко смоделированы. Программное обеспечение RoboAnalyzer имеет ряд преимуществ и несколько недостатков по сравнению с другим подобным ПО, как указано в работе [8].

В статье [9] предлагается методология разработки ПО для автономного и онлайн-моделирования пользовательского робота. Эта методология поможет производителю роботов использовать платформу RoboAnalyzer для разработки специального ПО для моделирования роботов.

Авторы статьи [10] считают, что повторное использование кода поощряется предоставлением облегченной платформы, а не требованием к пользователю модифицировать сложную систему.

В целом, для программирования роботов-манипуляторов с целью промышленной автоматизации необходимо указать желаемое движение и траекторию рабочего органа,

инструмента или датчика, прикрепленного к последнему звену манипулятора. Чтобы выполнить это условие, можно использовать различные методы, например кинематику, динамику, управление и оптимизацию, расчеты углов соединения и крутящих моментов, необходимых для достижения желаемого движения и траектории. Также можно использовать различные программные инструменты для взаимодействия с различными типами манипуляторов и датчиков роботов. На основе анализа современных программных средств и методологии разработки ПО, а также с учетом имеющегося в распоряжении у исследовательской группы авторов оборудования, была поставлена цель данного исследования: разработать ПО для управления роботом манипулятором на базе микроконтроллера STM32F407.

Результаты и обсуждение. Разработка ПО для управления роботом манипулятором на базе микроконтроллера STM32F407 производилась в четыре этапа, с выполнением обоснования по каждому этапу выбора оборудования оборудования и методов эксперимента.

1. *Разработка ПО в программе CUBE IDE.* Разработка ПО для управления роботами-манипуляторами, как упоминалось ранее, является сложной задачей. То есть, даже для простых операций захвата и размещения, разработчик ПО должен использовать аппаратный интерфейс, параллелизм, программирование в реальном времени, программирование сервоуправления, и генерация траектории в программную платформу. В более продвинутых приложениях манипуляторы работают на основе сенсорной и визуальной обратной связи. Наконец, многие современные системы оснащены усовершенствованными консолями оператора, основанными на визуальной обратной связи, виртуальной реальности и графическом пользовательском интерфейсе. Чтобы свести к минимуму усилия по разработке, желательно использовать существующую программную платформу. Однако разнообразие областей робототехнических исследований, приложений и роботизированного оборудования не способствовало развитию широко используемой платформы [11].

STM32Cube IDE – это встроенное ПО для разработки микроконтроллеров STM32 с конфигурацией периферийных устройств, генерацией кода, компиляцией кода и функциями отладки для микроконтроллеров и микропроцессоров STM32. STM32CubeIDE интегрирует функции конфигурации STM32 и создания проектов из STM32CubeMX, предлагая универсальный инструмент и экономя время на установку и разработку. После выбора пустого MCU или MPU STM32 или предварительно сконфигурированного микроконтроллера или микропроцессора из выбора платы или выбора примера создается проект и генерируется код инициализации. В любой момент разработки пользователь может вернуться к инициализации и настройке периферийных устройств или промежуточного программного обеспечения и повторно сгенерировать код инициализации, не влияя на пользовательский код [12].

В данной статье для считывания показаний датчика расстояния и передачи показаний на персональном компьютере используется устройство на базе микроконтроллера STM32F407 с интерфейсом RS485 и преобразователем USB-COM порт (рис. 1).

Семейство STM32F407xxb основано на высокопроизводительном 32-разрядном RISC-ядре ARM® Cortex™-M4, работающем на частоте до 168 МГц. Ядро Cortex-M4 оснащено модулем с плавающей запятой точности, который поддерживает все инструкции и типы данных обработки данных с одинарной точностью ARM. Он также реализует полный набор инструкций DSP и модуль защиты памяти, который повышает безопасность приложений [13].

Универсальный асинхронный приёмопередатчик (УАПП, англ. Universal Asynchronous

Receiver-Transmitter, UART) – узел вычислительных устройств, предназначенный для организации связи с другими цифровыми устройствами. Преобразует передаваемые данные в последовательный вид так, чтобы было возможно передать их по одной физической цифровой линии другому аналогичному устройству. Метод преобразования хорошо стандартизован и широко применяется в компьютерной технике (особенно во встраиваемых устройствах и системах на кристалле (анг. Security Operations Center (SoC)). Представляет собой логическую схему, с одной стороны подключённую к шине вычислительного устройства, а с другой имеющую два или более выводов для внешнего соединения [14].



Рисунок 1. Устройство на базе микроконтроллера STM32F407 с интерфейсом RS485 и преобразователем USB-COM порт

Шаги, выполняемые при разработке, приведены ниже:

- Конфигурируется интерфейс UART3 и ПИН PD10 на выход для управления микросхемой приемопередатчика RS485
- Конфигурируется интерфейс UART1 для создания COM-порта
- Подключается Serial Wire для отладки программы
- Конфигурируется блоки регистров RCC (анг. Reset and Clock Control) для установления внешнего генератора частоты
- Настройка частоты тактирования
- Ножка микроконтроллера PA12 назначается как «выход» (разрешение на перемещение руки).

На рис. 2 представлена разработанная схема для микроконтроллера, где показано как настраиваются частоты разной периферии, скорость работы, настройка частоты тактирования и т.д.

Основные рабочие линии у нас – RXD и TXD, или просто RX и TX. Передающая линия – TXD (анг. Transmitted Data), а порт RXD (анг. Received Data) – принимающая. Эти линии COM-порта задействованы при передаче без аппаратного управления потоком данных. Выход передатчика TX соединен с входом приемника RX и наоборот. Электрический принцип работы RS-232 отличается от стандартной 5-вольтовой TTL логики. В

этом протоколе логический ноль лежит от +3 до +12 вольт, а единица от -3 до -12, соответственно. Промежуток от -3 до +3 вольт считается зоной неопределенности.

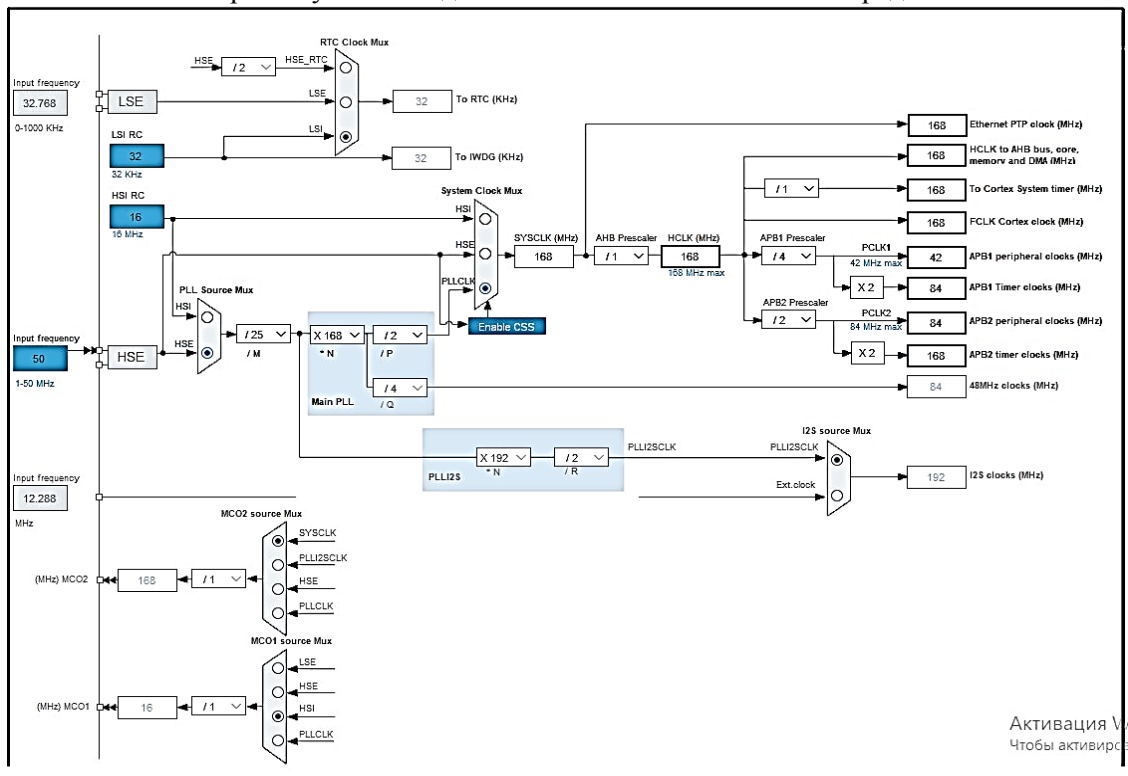


Рисунок 2. Схема для настройки параметров микроконтроллера

При настройке UART в данной статье подключается подтягивающий резистор на линию RX, выбирается необходимая скорость, настройки четности и количество стоповых битов. Разрешаются прерывания.

2. Программирование микроконтроллера

Линейки STM32F407/417 предназначены для промышленных, медицинских и потребительских приложений, где требуется высокий уровень интеграции и производительности, встроенная память и богатый набор периферийных устройств внутри корпусов размером всего 10 x 10 мм.

STM32F407/417 предлагает производительность ядра Cortex™-M4 (с модулем с плавающей запятой), работающего на частоте 168 МГц.

При программировании микроконтроллера были выполнены следующие шаги:

1. Создание глобальных переменных для управления пинами (программный код представлен ниже):

```

*/
/* USER CODE END Header */
/* Includes -----*/
#include "main.h"

/* Private includes -----*/
/* USER CODE BEGIN Includes */
#include <stdio.h>
    
```

```
#include <stdint.h>
#include <stdlib.h>
uint8_t rx_cmlt = 0;

#define TX_EN HAL_GPIO_WritePin(GPIOD, GPIO_PIN_10, SET)
#define RX_EN HAL_GPIO_WritePin(GPIOD, GPIO_PIN_10, RESET)

#define DO_ON
#define DO_OFF

uint8_t rx_cmlt_COM;
uint8_t mode = 0;
uint8_t rx_data_RTU[50];
uint8_t tx_data_RTU[] = {0xff, 20, 30, 50};
uint8_t tx_data_COM[10];
uint8_t len_message = 4;
uint8_t len_message_COM = 4;
uint8_t ready = 1;
/* USER CODE END Includes */

/* Private typedef -----*/
/* USER CODE BEGIN PTD */

/* USER CODE END PTD */

/* Private define -----*/
/* USER CODE BEGIN PD */
/* USER CODE END PD */

/* Private macro -----*/
/* USER CODE BEGIN PM */

/* USER CODE END PM */

/* Private variables -----*/
I2C_HandleTypeDef hi2c3;

SPI_HandleTypeDef hspi1;
SPI_HandleTypeDef hspi3;

TIM_HandleTypeDef htim1;
TIM_HandleTypeDef htim2;
TIM_HandleTypeDef htim5;
TIM_HandleTypeDef htim9;
TIM_HandleTypeDef htim12;

UART_HandleTypeDef huart1;
UART_HandleTypeDef huart3;
```

```
/* USER CODE BEGIN PV */
/* USER CODE END PV */

/* Private function prototypes -----*/
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_USART3_UART_Init(void);
static void MX_TIM1_Init(void);
static void MX_USART1_UART_Init(void);
static void MX_SPI1_Init(void);
static void MX_SPI3_Init(void);
static void MX_TIM2_Init(void);
static void MX_TIM12_Init(void);
static void MX_TIM9_Init(void);
static void MX_TIM5_Init(void);
static void MX_I2C3_Init(void);
/* USER CODE BEGIN PFP */

/* USER CODE END PFP */

2. Основной цикл программы (программный код представлен ниже):
Modbus_init(9600,1,1);
HAL_TIM_Base_Start_IT(&htim9);
ready = 1;
DO_OFF;
TX_EN;
HAL_UART_Transmit(&huart3, tx_data_RTU, len_message, 1000);
HAL_GPIO_TogglePin(GPIOF, LED_3_Pin);
mode = 0;
RX_EN;
HAL_UARTEx_ReceiveToIdle_IT(&huart3, rx_data_RTU, 50);

/* USER CODE END 2 */

/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
/* USER CODE END WHILE */

/* USER CODE BEGIN 3 */
if(rx_cmlt == 1){
ready = 0;
mode = 1;
HAL_GPIO_TogglePin(GPIOF, LED_4_Pin);
rx_cmlt = 0;
HAL_UART_Transmit(&huart1, tx_data_COM, len_message_COM, 1000);
```



```
        ready = 1;
    }
    if(ready == 1){
        if(mode == 1){DO_ON;HAL_Delay(100);DO_OFF;
        HAL_Delay(2000);TX_EN;HAL_UART_Transmit(&huart3, tx_data_RTU, len_message,
        1000);HAL_GPIO_TogglePin(GPIOF, LED_3_Pin); mode = 0;RX_EN;}
        if(mode == 0){HAL_UARTEx_ReceiveToIdle_IT(&huart3, rx_data_RTU, 50);}}
```

В цикле While мы переключаем порт микроконтроллера (выход) управления приемопередатчиком в режим отправки и отправляем запрос датчику по шине RS485 (протокол Modbus RTU) запрос на получения результата измерения.

В основе интерфейса RS-485 лежит принцип дифференциальной (балансной) передачи данных. Суть его заключается в передаче одного сигнала по двум проводам. Причем по одному проводу (условно А) идет оригинальный сигнал, а по другому (условно В) — его инверсная копия. Другими словами, если на одном проводе «1», то на другом «0» и наоборот. Таким образом, между двумя проводами витой пары всегда есть разность потенциалов: при «1» она положительна, при «0» — отрицательна. Именно этой разностью потенциалов и передается сигнал. Такой способ передачи обеспечивает высокую устойчивость к синфазной помехе. Синфазной называют помеху, действующую на оба провода линии одинаково. К примеру, электромагнитная волна, проходя через участок линии связи, наводит в обоих проводах потенциал. Если сигнал передается потенциалом в одном проводе относительно общего, как в RS-232, то наводка на этот провод может исказить сигнал относительно хорошо поглощающего наводки общего («земли»). Кроме того, на сопротивлении длинного общего провода будет падать разность потенциалов земель - дополнительный источник искажений. А при дифференциальной передаче искажения не происходит. В самом деле, если два провода пролегают близко друг к другу, да еще перевиты, то наводка на оба провода одинакова. Потенциал в обоих одинаково нагруженных проводах изменяется одинаково, при этом информативная разность потенциалов остается без изменений.

Переключаем порт микроконтроллера (выход) управления приемопередатчиком в режим приема и ждем данных от датчика.

При успешном приеме данных мы инкрементируем переменную счета запросов, копируем данных с датчика в основной буфер, затираем буфер приема и переключаем порт разрешения движения руки, ждем 1мс и сбрасываем.

Ждем 5с и повторяем алгоритм.

После прохождения 100 точек отправляем данные в СОМ-порт.

3. Программирование в *Ciros Programming*

CIROS - это испытанная в промышленных масштабах чрезвычайно мощная платформа для создания и использования трехмерных имитационных моделей для технологий автоматизации. CIROS, дополненный дидактическими механизмами и материалами, можно использовать различными способами в учебной среде [15].

Одной из важных возможности CIROS является наличие симуляций ошибок, для представления возможных вариантов при совершении некорректной работы системы. Можно организовать цветовую индикацию, анимацию ошибки и в виде данных в таблице. Приложение поддерживает симуляцию физики во время исполнения. Для каждого компонента можно задать определенные параметры, такие как вид материала, массу, может ли предмет используя кинетическое движение, плотность и другие. Все это можно оценить при помощи доступных измерений.

Для программирования в *Ciros Programming* выполнены следующие шаги:

1. Создание карты точек движения манипулятора: 100 точек с шагом 10 мм на 10 мм.

Ширина и Длина поля сканирования 100 мм на 100 мм.

2. Написание программы движения манипулятора

При появлении дискретного сигнала, манипулятор передвигается на следующую точку и делает задержку в 1 секунду.

4. *Обработка данных в приложении*

Программа для визуализации данных представлена на рисунке 3.

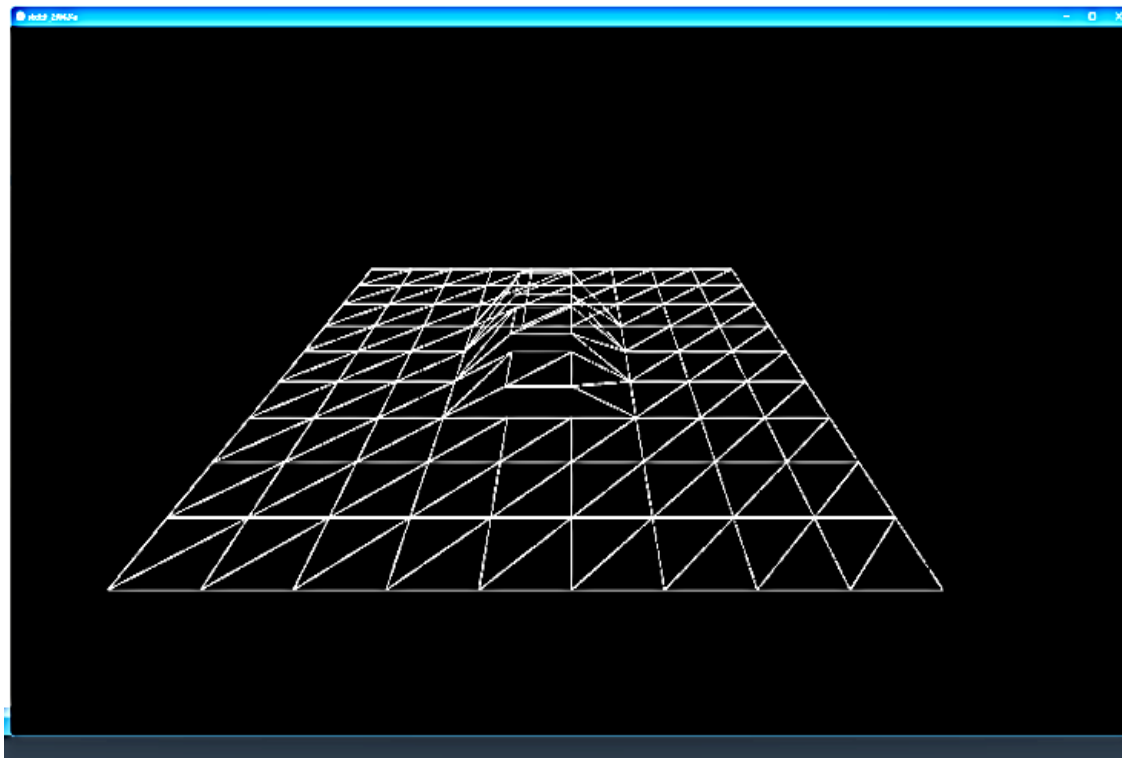


Рисунок 3. Визуализация полученных данных

Приложение подключается к COM-порту и визуализирует полученные данные.

Заключение. Роботы-манипуляторы могут повысить производительность, качество, безопасность и гибкость производства в промышленных условиях, поскольку они могут работать быстрее, точнее и надежнее, чем люди. Для программирования роботов-манипуляторов для промышленной автоматизации необходимо указать желаемое движение и траекторию рабочего органа, инструмента или датчика, прикрепленного к последнему звену манипулятора. Для считывания показаний датчика расстояния и передачи показаний на ПК используется устройство на базе микроконтроллера STM32F407 с интерфейсом RS485 и преобразователем USB-COM порт и робота-манипулятора. Разработка программного обеспечения производилась в четыре этапа: программирование в программе CUBE IDE, программирование микроконтроллера, программирование в Ciroc Programming, обработка данных в приложении. ПО разработано на основе имеющегося у исследовательской группы авторов оборудования, которое имеет такие преимущества, как простота в разработке и дешевизна, а также доступность программных инструментов для взаимодействия с различными типами

манипуляторов и датчиков. Главным разработчиком данного ПО является инженер-программист исследовательской группы Токмин М.

Благодарность. Данное исследование финансируется Комитетом науки Министерства науки и высшего образования Республики Казахстан (грант № AP13068317«Разработка новых алгоритмов управления роботом – манипулятором для технологий 3D сканирования и аддитивного микроплазменного напыления покрытий»).

Список литературы

1. Vaibhav Gupta, Rajeevlochana G. Chittawadigi, S. Saha. RoboAnalyzer: Robot Visualization Software for Robot Technicians. Advanced Robotics, 2017, pp. 1-5.
 2. A. M. Moshi.A, S. S. Cynthia, E. Islam, R. Rahman, A. K. M. Abdul Malek Azad. Performance analysis of robotic arm manipulators control system under multitasking environment. IEEE, 2011 DOI: 10.1109/ICIEEM.2011.6035233.
 3. Othayoth, Ratan S., Chittawadigi, Rajeevlochana G., Joshi, Ravi P., and Saha, Subir K. Robot kinematics made easy using RoboAnalyzer software. Computer Applications in Engineering Education, 2017, pp. 669-680.
 4. Z. Li, L. Huang, X. Wu. Real-time Dynamic Information Display Modes of Pharmaceutical Robots with Multitasking Interaction. Conference: HCI International 2023, 25th International conference on human-computer interaction, 2022.
 5. G. Cai, J. Gong, S.Peng, W. Wei. Finite Element Model of Kinematic Pair Clearances of Elastic Linkage Mechanisms. Journal of Physics Conference Series 2449(1):012018, 2023.
 6. A. M. Titu, V. Gusan, A. B. Pop. Methods and Techniques Utilized in Programming Collaborative Robots for High-Quality Automation. In book: International Conference on Reliable Systems Engineering (ICoRSE), 2023
 7. Saha, S. Kumar. Introduction to Robotics. Second edition, Tata McGraw Hill Publications, 2014.
 8. Rajeevlochana, C. G., Saha, Subir K, Kumar, Shivesh. Automatic extraction of DH parameters of serial manipulators using linegeometry. In Proceedings of the 2nd International Conference on Multibody System Dynamics, 2012.
 9. Sadanand, Ratan., Chittawadigi, Rajeevlochana G., Joshi, Ravi P., and Saha, Subir K. Virtual robots module: an effective visualization tool for robotics toolbox. In Proceedings of the 2nd International Conference on Advances in Robotics, 2015.
 10. S. Kozibroda, A. Palchyk, I.- S. Mazur V. Rak Remote learning system model for robot programming. Published by Ternopil Volodymyr Hnatiuk National Pedagogical University, 2023
 11. STM32CubeIDE Integrated Development Environment for STM32 [Электронный ресурс] Режим доступа: <https://www.st.com/en/development-tools/stm32cubeide.html>
 12. STM32F407ze Documentation. [Электронный ресурс] Режим доступа: <https://www.st.com/en/microcontrollers-microprocessors/stm32f407ze.html>
 13. A.K. Gupta, A. Raman, N. Kumar, R. Ranjan. Design and Implementation of High-Speed Universal Asynchronous Receiver and Transmitter (UART). Conference: 2020 7th International Conference on Signal Processing and Integrated Networks (SPIN), 2020.
 14. 32-Bit FLASH 512KB (512K x 8) Microcontroller [Электронный ресурс] Режим доступа: https://www.kynix.com/Detail/413712/STM32F407ZET6.html?utm_source=google&utm_medium=pc&utm_campaign=PMax&gad=1&gclid=CjwKCAjwJimBhAsEiwA1hrp5IUxaNIMeZeDthd6JySWw7zULRQoRpSqePSuGBDixUp7s47meBhPshoCusEQAvD_BwE
 15. S. Nickel, C. Steinhardt, H. Steinhardt, W. Burkart. Selected features of OPL and CPLEX Optimization Studio. In book: Decision Optimization with IBM ILOG CPLEX Optimization Studio, A Hands-On Introduction to Modeling with the Optimization Programming Language (OPL), 2022.
-
-