

Трубин Денис Александрович

Бакалавр, СПбГАСУ, Санкт-Петербург

РАЗРАБОТКА ПРОГРАММНОГО МОДУЛЯ ПРЯМОГО УПРАВЛЕНИЯ ПЛАТОЙ ЛАЗЕРНОЙ МАРКИРОВКИ ЧЕРЕЗ ДИНАМИЧЕСКУЮ БИБЛИОТЕКУ

Аннотация. В статье рассматривается разработка автономного программного модуля для прямого управления платой лазерной маркировки с использованием динамических библиотек штатного программного окружения. Актуальность работы обусловлена необходимостью автоматизации процессов маркировки без интерактивного использования пользовательского интерфейса штатной программы. Описана общая архитектура модуля, включающая обработку входных данных, формирование задания, настройку технологических параметров, загрузку управляющей библиотеки, инициализацию платы и контроль выполнения. Особое внимание уделено командному интерфейсу, режиму предварительного просмотра, ведению журнала выполнения и безопасному запуску маркировки. Для проверки работоспособности предложена методика экспериментальной проверки на тестовых геометрических заданиях и синтетических двумерных кодах, не содержащих производственной или коммерческой информации. Полученные результаты могут быть использованы при разработке программно-аппаратных систем автоматизированной лазерной маркировки.

Ключевые слова: лазерная маркировка, динамическая библиотека, программно-аппаратный модуль, плата управления, командный интерфейс, автоматизация, Data Matrix.

Trubin Denis Aleksandrovich

Bachelor, SPbGASU, Saint Petersburg

DEVELOPMENT OF A SOFTWARE MODULE FOR DIRECT CONTROL OF A LASER MARKING BOARD THROUGH A DYNAMIC LIBRARY

Annotation. The article discusses the development of an autonomous software module for direct control of a laser marking board using dynamic libraries of the standard software environment. The relevance of the work is determined by the need to automate marking processes without interactive use of the standard graphical user interface. The paper describes the general architecture of the module, including input data processing, task generation, technological parameter configuration, loading of the control library, board initialization, and execution control. Special attention is paid to the command-line interface, preview mode, execution logging, and safe start of marking operations. The experimental verification methodology is based on test geometric tasks and synthetic two-dimensional codes that do not contain production or commercial information. The results can be used in the development of hardware-software systems for automated laser marking.

Keywords: laser marking, dynamic library, hardware-software module, control board, command-line interface, automation, Data Matrix.

Введение

Лазерная маркировка применяется для нанесения серийных номеров, штрихкодов, двумерных кодов и другой идентификационной информации на изделия [1, 2]. Обычно управление маркиратором выполняется через штатное программное обеспечение производителя, где оператор вручную открывает макет, задает параметры и запускает процесс. Такой вариант удобен для разовых операций, но плохо подходит для автоматизации, когда задания должны формироваться и запускаться из внешней программы.

В современных исследованиях лазерная маркировка рассматривается не только как способ нанесения символов, но и как элемент автоматизированных производственных систем. Отмечается, что при нанесении переменной информации и двумерных кодов важную роль играют не только параметры лазерного воздействия, но и система управления, обеспечивающая формирование задания, синхронизацию операций и контроль выполнения [3]. В работах по автоматизации лазерной маркировки также рассматриваются подходы с применением машинного зрения и роботизированных систем, что подтверждает актуальность перехода от ручного управления к программно управляемым комплексам [4]. Отдельное внимание в литературе уделяется влиянию мощности, скорости, частоты и материала заготовки на качество маркировки [5]. Для задач автоматической идентификации изделий актуально использование двумерных кодов Data Matrix, позволяющих размещать информацию на ограниченной площади [6].

В работе рассматривается разработка автономного программного модуля, который управляет платой лазерной маркировки через динамические библиотеки штатного программного окружения. Модуль получает входные данные, формирует задание,

задает технологические параметры, инициализирует плату и запускает маркировку без обязательного использования графического интерфейса штатной программы.

Цель работы - разработать программный модуль прямого управления платой лазерной маркировки через динамическую библиотеку.

Для достижения цели были поставлены следующие задачи: определить общую схему взаимодействия программы с платой управления; реализовать загрузку управляющих библиотек и инициализацию оборудования; разработать командный интерфейс для передачи параметров маркировки; реализовать формирование задания и режим предварительного просмотра; проверить работу модуля на тестовых заданиях.

1. Архитектура программного модуля

Разрабатываемый программный модуль предназначен для запуска заданий лазерной маркировки без интерактивного использования штатного графического интерфейса. Общая логика работы модуля основана на последовательной обработке входных данных, формировании геометрического задания, передаче технологических параметров и обращении к плате управления через динамические библиотеки штатного программного окружения.

Структурно модуль можно разделить на несколько основных частей: блок чтения входных данных, парсер задания, генератор геометрии, блок технологических параметров, программную обертку над динамической библиотекой, блок инициализации платы, блок запуска маркировки и систему ведения журнала выполнения. Такая структура позволяет отделить подготовку задания от аппаратного запуска, что упрощает отладку и снижает риск ошибочной маркировки. Обобщенная схема работы программного модуля представлена на рисунке 1.

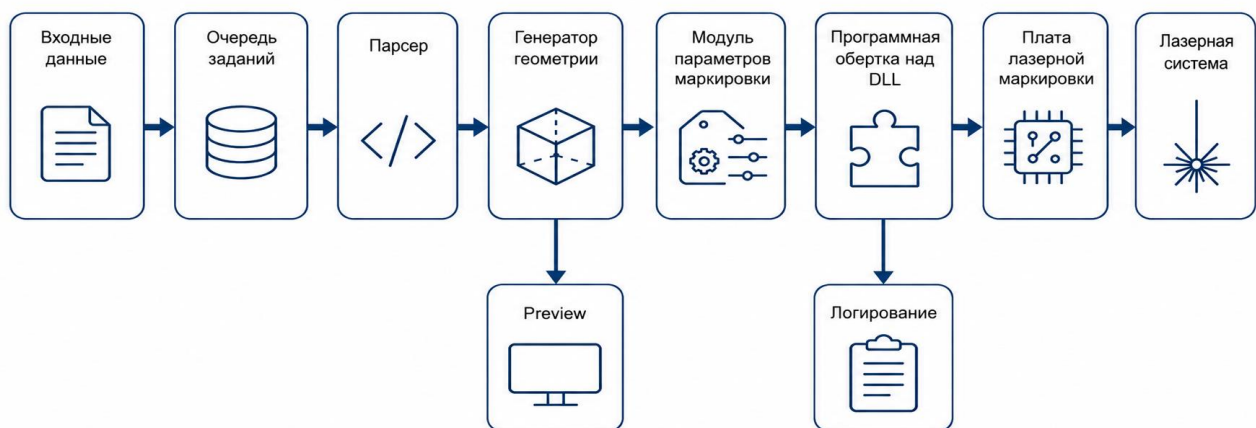


Рисунок 1 - Архитектура программного модуля

Входные данные могут передаваться в виде файла очереди или параметров командной строки. После чтения данных модуль выполняет их проверку и преобразует в промежуточное представление. Далее генератор геометрии формирует набор примитивов, пригодных для передачи в управляющий слой. Отдельный блок отвечает за технологические параметры маркировки: мощность, скорость, частоту, шаг заливки, координаты начала задания и режим выполнения.

Связь с оборудованием выполняется через программную обертку над динамическими библиотеками. Обертка скрывает низкоуровневые вызовы и предоставляет модулю единую последовательность действий: загрузка библиотеки, инициализация платы, передача параметров, запуск задания, ожидание завершения и освобождение ресурсов. Такой подход делает модуль более удобным для дальнейшего расширения и позволяет использовать его как основу для автоматизированной системы маркировки.

Дополнительно в архитектуре предусмотрен режим предварительного просмотра. В этом режиме задание формируется полностью, но аппаратный запуск не выполняется. Это позволяет проверить корректность геометрии и параметров до включения лазера. Для контроля работы модуль также ведет журнал выполнения, в котором фиксируются загруженные задания, параметры запуска, коды возврата и результат выполнения операции.

2. Динамическая загрузка библиотек и инициализация платы

Взаимодействие программного модуля с лазерной маркировочной системой выполняется через динамические библиотеки штатного программного окружения [7]. Такой подход позволяет не обращаться к пользовательскому интерфейсу исходной программы, а выполнять необходимые операции из собственного приложения: загрузку библиотеки, поиск платы управления, инициализацию оборудования, подготовку задания и запуск маркировки.

На этапе запуска модуль устанавливает рабочий каталог, загружает управляющие библиотеки и получает адреса необходимых функций [8, 9]. После успешной загрузки выполняется открытие драйвера, поиск доступных устройств и выбор активной платы лазерной маркировки. Если устройство не обнаружено или библиотека не загружена, выполнение задания прекращается, а информация об ошибке записывается в журнал.

После обнаружения платы выполняется ее инициализация и подготовка к маркировке. На этом этапе модуль переводит оборудование в рабочее состояние, задает контекст управления, проверяет готовность к выполнению задания и передает технологические параметры. Далее сформированное геометрическое задание передается в управляющий слой, после чего выполняется запуск маркировки и ожидание завершения операции.

Обобщенная последовательность работы модуля представлена в таблице 1.

Таблица 1 - Основные этапы взаимодействия программного модуля с платой лазерной маркировки

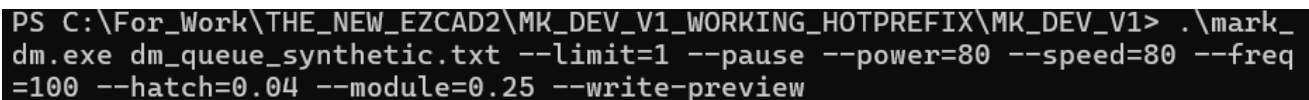
Этап	Назначение	Результат
Загрузка библиотек	Подключение управляющих DLL	Библиотеки загружены
Получение функций	Подготовка вызовов управляющего API	Функции доступны
Поиск устройства	Определение платы лазерной маркировки	Найдено одно устройство
Инициализация платы	Подготовка оборудования к работе	Инициализация выполнена
Передача параметров	Настройка режима маркировки	Параметры применены
Запуск задания	Передача геометрии и запуск маркировки	Задание выполнено
Завершение работы	Закрытие драйвера и освобождение ресурсов	Ресурсы освобождены

После завершения маркировки модуль проверяет код возврата, фиксирует результат выполнения в журнале и освобождает занятые ресурсы. Завершение работы включает закрытие драйвера, отвязку контекста управления и выгрузку динамических библиотек. Такая последовательность позволяет обеспечить повторяемый запуск заданий и снизить вероятность ошибок, связанных с ручным управлением через штатный графический интерфейс.

3. Командный интерфейс и входные данные

Управление программным модулем выполняется через командный интерфейс. Такой вариант выбран, поскольку он удобен для автоматизации: программу можно запускать из командной строки, пакетных файлов, скриптов или внешних информационных систем. В отличие от графического интерфейса, командный запуск позволяет заранее задать все необходимые параметры и исключить ручную настройку перед каждым выполнением задания.

В качестве входных данных модуль принимает файл очереди заданий и набор параметров командной строки. Файл очереди содержит одно или несколько заданий маркировки, а параметры командной строки определяют режим выполнения, технологические параметры лазера и настройки формирования геометрии. Пример запуска модуля через командный интерфейс представлен на рисунке 2.



```
PS C:\For_Work\THE_NEW_EZCAD2\MK_DEV_V1_WORKING_HOTPREFIX\MK_DEV_V1> .\mark_
dm.exe dm_queue_synthetic.txt --limit=1 --pause --power=80 --speed=80 --freq
=100 --hatch=0.04 --module=0.25 --write-preview
```

Рисунок 2 - Пример запуска программного модуля через командный интерфейс

Параметр `--limit` задает количество заданий, которые необходимо обработать из очереди. Режим `--pause` включает ручное подтверждение перед запуском маркировки, что удобно при установке детали оператором. Режим `--auto` предназначен для последовательного выполнения заданий без дополнительного подтверждения. Параметр `--preview` позволяет сформировать предварительное представление задания без запуска лазера.

Технологические параметры задаются отдельными ключами командной строки. К ним относятся мощность лазера, скорость маркировки, частота импульсов, шаг заливки, координаты начала задания и размер элемента формируемой геометрии.

За счет этого один и тот же модуль может использоваться для разных режимов обработки без изменения исходного кода программы. Основные параметры командного интерфейса приведены в таблице 2.

Таблица 2 - Основные параметры командного интерфейса

Параметр	Назначение
dm_queue_synthetic.txt	Входной файл с очередью заданий
--limit	Ограничение количества заданий
--pause	Ручное подтверждение перед запуском
--auto	Автоматическое выполнение заданий
--preview	Предпросмотр без запуска лазера
--write-preview	Сохранение файлов предпросмотра
--power	Мощность лазера, %
--speed	Скорость маркировки, мм/с
--freq	Частота импульсов, кГц
--hatch	Шаг заливки, мм
--x, --y	Координаты начала задания
--module	Размер модуля кода, мм
--fill	Режим заливки
--delay-ms	Задержка между заданиями

Использование файла очереди позволяет отделить подготовку данных от процесса маркировки. В экспериментальной части работы применялась синтетическая очередь заданий, не содержащая производственной или коммерческой информации. Такой подход дает возможность проверить работу парсера, генератора геометрии и управляющего слоя без раскрытия реальных кодов маркировки.

4. Формирование задания маркировки

Формирование задания маркировки выполняется после чтения входного файла и параметров командной строки. На этом этапе модуль преобразует исходные данные в набор геометрических примитивов, которые затем передаются в управляющий слой для выполнения маркировки. Такой подход позволяет отделить логическую подготовку задания от аппаратного запуска лазерной системы.

Входной файл очереди содержит одно или несколько заданий. Каждое задание включает идентификатор, служебное имя и блок данных, описывающий двумерную матрицу. В экспериментальной проверке использовалась синтетическая последовательность, не относящаяся к производственным кодам маркировки. Это позволило проверить работу алгоритма без раскрытия конфиденциальной информации. Фрагмент входного файла очереди заданий представлен на рисунке 3.

```
BEGIN 9001 synthetic_dm_article_test
# synthetic test sequence, no production or commercial data
# rows=22 cols=22
DATA
11111111111111111111111111
1000001000100010000001
1011101011101010111101
1010101000100010100101
...
END
```

Рисунок 3 - Фрагмент входного файла очереди заданий

После чтения задания модуль выполняет разбор входных строк, проверяет структуру блока данных и формирует внутреннее представление матрицы. Далее элементы матрицы преобразуются в геометрические области с учетом заданного

размера модуля, начальных координат и шага заливки. Черные элементы двумерного кода преобразуются в набор линий или прямоугольных областей, которые затем передаются в виде последовательности команд перемещения и маркировки.

Общий алгоритм формирования задания включает последовательное чтение очереди, выделение блока данных, проверку структуры матрицы, построение внутреннего представления, преобразование элементов матрицы в геометрические области, применение технологических параметров, передачу команд перемещения и маркировки в управляющий слой, ожидание завершения операции и запись результата в журнал выполнения.

Перед аппаратным запуском задание может быть сохранено в режиме предварительного просмотра. Это позволяет проверить корректность построенной геометрии без включения лазера. После успешного формирования и проверки задания модуль передает его в управляющий слой, где выполняется настройка параметров, запуск маркировки и контроль завершения операции.

5. Режим предварительного просмотра и технологические параметры

Для снижения риска ошибочной маркировки в программном модуле предусмотрен режим предварительного просмотра. В этом режиме входное задание полностью обрабатывается: выполняется чтение очереди, разбор данных, построение геометрии и применение заданных параметров. При этом аппаратный запуск лазера не выполняется, а результат сохраняется в виде файлов предварительного просмотра.

Режим предварительного просмотра позволяет проверить расположение элементов, размеры формируемой геометрии и общую структуру задания до передачи команд на плату управления. Это особенно важно при работе с двумерными кодами, так как ошибка в размере модуля, координатах или шаге заливки может привести к некорректной маркировке. В рамках работы использовались текстовое представление задания и SVG-файл, позволяющий визуально оценить результат построения. Пример предварительного просмотра синтетического двумерного кода представлен на рисунке 4.

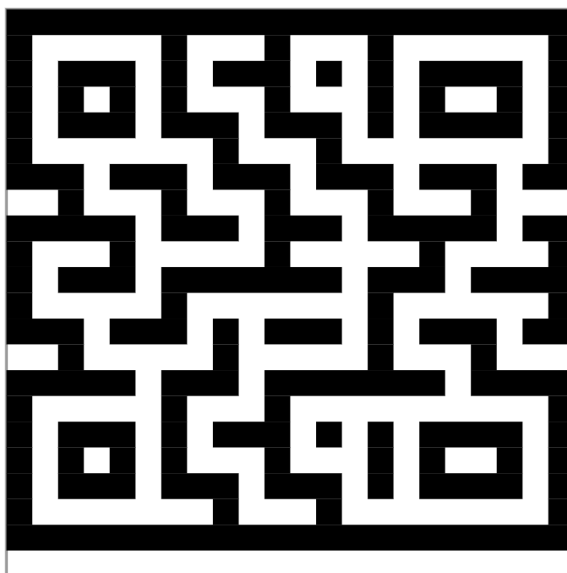


Рисунок 4 - Предварительный просмотр синтетического двумерного кода

Помимо формирования геометрии, модуль позволяет задавать технологические параметры маркировки. К основным параметрам относятся мощность лазера, скорость маркировки, частота импульсов, шаг заливки, координаты начала задания и режим выполнения очереди [10]. Также могут использоваться параметры задержек, влияющие на включение и выключение лазерного излучения при переходах между элементами задания. Основные группы параметров приведены в таблице 3.

Таблица 3 - Группы технологических и управляющих параметров

Группа параметров	Параметры
Геометрия задания	x, y, module, hatch, fill
Параметры лазера	power, speed, freq
Задержки	laser on delay, laser off delay, mark end delay, polygon delay
Режим очереди	limit, pause, auto, delay-ms
Предварительный просмотр	preview, write-preview
Калибровка	scale_x/y, shear_x/y, rotate, offset_x/y, mirror_x/y

Разделение параметров на группы упрощает настройку процесса маркировки. Геометрические параметры определяют положение и размеры задания, технологические параметры влияют на характер воздействия лазера на материал, а управляющие параметры задают режим выполнения очереди. Такой подход позволяет использовать один программный модуль для разных тестовых режимов без изменения исходного кода.

6. Экспериментальная проверка

Для проверки работоспособности программного модуля были проведены тестовые запуски на синтетических данных, не содержащих производственной или коммерческой информации. В качестве основного задания использовался тестовый двумерный код, не относящийся к реальным производственным кодам маркировки. Такой подход позволил проверить работу модуля без раскрытия реальных кодов маркировки.

На первом этапе была проверена инициализация оборудования. Модуль выполнил загрузку управляющих библиотек, обнаружил одно подключенное устройство, выполнил инициализацию платы лазерной маркировки и перешел в режим готовности к выполнению задания. Это подтвердило корректность базовой последовательности: загрузка библиотек, поиск устройства, выбор активной платы и подготовка к маркировке.

На втором этапе был выполнен режим предварительного просмотра. Программа обработала входной файл очереди, сформировала внутреннее представление задания и сохранила результат в виде SVG/TXT-файлов. Аппаратный запуск лазера на данном этапе не выполнялся. Это позволило проверить структуру задания и корректность формирования геометрии до передачи команд на плату управления.

На третьем этапе был выполнен полный цикл маркировки синтетического двумерного кода. В ходе теста программа считала одно задание из очереди, применила заданные технологические параметры, сформировала набор линий заливки, передала задание на плату управления и дождалась завершения операции. По результатам выполнения был получен код возврата $gc=0$, что соответствует успешному завершению маркировки. Фрагмент консольного журнала выполнения полного цикла маркировки представлен на рисунке 5.

```

mark_dm autonomous DM marker
queue=dm_queue_synthetic.txt limit=1 previewOnly=0 writePreview=1 pause=1 fill=horz origin=0.000 0.000 moduleX=0.250 moduleY=0.250 xscale=1.0000 yscale=1.0000 hatch=0.040 power=80.0 speed=80.0 freq=100 tc(on/off/end/poly)=0/100/300/100 delayMs=0
loaded queue: 1 codes from dm_queue_synthetic.txt
preview 1: preview_dm\dm_0001_id9001.txt preview_dm\dm_0001_id9001.svg
LoadLibrary(DataMgr.dll) -> 10000000 err=0
LoadLibrary(Lmc1_real.dll) -> 026C0000 err=0
lmc_OpenDriver(0) -> 1
lmc_GetValidDev -> rc=0 count=1 dev=02CD0048
InitGetBoardHardInfo -> 1
InitLmc(0,0) -> 0
[1/1] Put next part for code id=9001, then press ENTER to mark...MARK DM id=9001 name=synthetic_dm_article_test
DM id=9001 commands: fill=horz hatch lines=784 size=22x22 moduleX=0.250mm moduleY=0.250mm xscale=1.0000 yscale=1.0000 hatch=0.040mm
MARK DONE id=9001 rc=0
CODE id=9001 total mark time: 1375 ms
lmc_CloseDriver done
done

```

Рисунок 5 - Фрагмент консольного журнала выполнения полного цикла маркировки синтетического двумерного кода

Основные параметры тестового запуска приведены в таблице 4.

Таблица 4 - Параметры тестового запуска

Параметр	Значение
Тип задания	синтетический двумерный код
Размер матрицы	22×22 элемента
Размер модуля	0,25 мм
Шаг заливки	0,04 мм
Количество линий заливки	784
Мощность лазера	80 %
Скорость маркировки	80 мм/с
Частота	100 кГц
Время выполнения задания	1375 мс
Код завершения	rc=0

В ходе проверки использовались следующие значения: размер матрицы - 22×22 элемента, размер модуля - 0,25 мм, шаг заливки - 0,04 мм, количество сформированных линий заливки - 784. Мощность лазера составляла 80 %, скорость маркировки - 80 мм/с, частота - 100 кГц. Время выполнения задания составило 1375 мс. Результат маркировки синтетического двумерного кода на тестовом образце представлен на рисунке 6.



Рисунок 6 - Результат маркировки синтетического двумерного кода на тестовом образце

Дополнительно была проведена проверка влияния технологических параметров на выполнение задания. Для этого однотипное задание запускалось в трех режимах с различными значениями мощности, скорости и шага заливки. Результаты проверки приведены в таблице 5.

Таблица 5 - Сравнение режимов маркировки

Режим	Мощность, %	Скорость, мм/с	Шаг заливки, мм	Линии заливки	Время, мс
А	40	120	0,05	672	1187
В	70	80	0,04	784	1360
С	100	40	0,03	1008	1734

При уменьшении шага заливки количество формируемых линий увеличивалось, что приводило к росту объема управляющего задания. Так, при шаге 0,05 мм было сформировано 672 линии, при шаге 0,04 мм - 784 линии, а при шаге 0,03 мм - 1008 линий. Время выполнения задания увеличилось с 1187 до 1734 мс.

Следует отметить, что время выполнения зависит не только от количества линий заливки. На него также влияют накладные расходы на передачу команд в управляющий слой, служебные вызовы управляющего API, ожидание завершения операции и особенности взаимодействия с платой управления. Поэтому зависимость времени от количества линий не является строго линейной, однако общая тенденция сохраняется: уменьшение шага заливки увеличивает объем задания и длительность маркировки.

Это подтверждает, что параметры, передаваемые через командный интерфейс, влияют как на структуру формируемого задания, так и на длительность его выполнения. В частности, параметр `hatch` изменяет количество линий заливки, а технологические параметры мощности и скорости определяют режим физического воздействия лазера на материал. Сравнение результатов маркировки при различных технологических параметрах представлено на рисунке 7.



Рисунок 7 - Сравнение результатов маркировки при различных технологических параметрах

Проведенные тесты подтвердили работоспособность основных функций программного модуля: загрузку библиотек, инициализацию платы, обработку входной очереди, формирование задания, режим предварительного просмотра, передачу задания на оборудование, запуск маркировки и фиксацию результата в журнале выполнения.

7. Безопасность и ограничения

При разработке программного модуля учитывалось, что лазерная маркировка является аппаратным процессом, связанным с воздействием на материал. Поэтому в модуле предусмотрены средства, снижающие риск ошибочного запуска задания. К таким средствам относятся режим предварительного просмотра, ограничение количества обрабатываемых заданий, ручное подтверждение перед маркировкой и фиксация результата выполнения в журнале.

Режим предварительного просмотра позволяет сформировать задание без включения лазера и проверить его до аппаратного запуска. Ограничение количества заданий через параметр командной строки предотвращает случайную обработку всей очереди. Режим ручного подтверждения используется для контроля момента запуска, когда оператор должен установить образец и подтвердить готовность к маркировке.

Отдельное ограничение связано с защитой производственных данных. В статье не приводятся реальные коды маркировки, клиентские данные и сведения, относящиеся к реальным производственным системам маркировки. Экспериментальная проверка выполнялась на синтетических данных, не содержащих коммерческой или производственной информации.

Также в работе не рассматриваются внутренние механизмы закрытого программного обеспечения производителя. Описание ограничено архитектурой разработанного модуля, последовательностью взаимодействия с динамическими библиотеками, формированием задания и результатами экспериментальной проверки. Такой подход позволяет показать принцип работы программного решения без раскрытия закрытых деталей штатного программного окружения.

К ограничениям текущей реализации можно отнести отсутствие полноценного графического интерфейса оператора и необходимость предварительной настройки рабочей директории с требуемыми библиотеками и конфигурационными файлами. Кроме того, качество маркировки зависит от выбранных технологических параметров, материала образца и корректности формирования геометрии задания. Дальнейшее развитие модуля может быть связано с расширением алгоритмов генерации двумерных кодов, автоматической калибровкой геометрии и интеграцией с внешними производственными системами.

Заключение

В результате работы был разработан автономный программный модуль прямого управления платой лазерной маркировки через динамические библиотеки штатного программного окружения. Модуль позволяет выполнять загрузку управляющих библиотек, инициализацию платы, обработку входной очереди, настройку технологических параметров, формирование задания маркировки и запуск процесса без обязательного использования графического интерфейса штатной программы.

В разработанном решении реализован командный интерфейс, позволяющий задавать режим выполнения, количество заданий, координаты, мощность, скорость, частоту, шаг заливки и другие параметры. Также предусмотрен режим предварительного просмотра, который позволяет проверить сформированное задание до аппаратного запуска лазера. Это снижает риск ошибочной маркировки и упрощает отладку задания.

Экспериментальная проверка показала, что модуль обеспечивает полный цикл работы: от чтения входных данных и формирования геометрии до передачи задания на плату управления и фиксации результата выполнения. При проверке на синтетическом двумерном коде маркировка была завершена с кодом возврата $rc=0$, что подтверждает корректность базовой последовательности управления.

Практическая значимость работы заключается в возможности использования разработанного подхода как основы для автоматизации лазерной маркировки. В дальнейшем модуль может быть расширен за счет развития алгоритмов формирования двумерных кодов, программной коррекции геометрических искажений, добавления операторского интерфейса и интеграции с внешними производственными системами.

Список литературы

1. ГОСТ Р ИСО/МЭК 16022-2008. Автоматическая идентификация. Кодирование штриховое. Спецификация символики штрихового кода Data Matrix.
2. ГОСТ Р ИСО/МЭК 19762-2-2011. Информационные технологии. Технологии автоматической идентификации и сбора данных. Гармонизированный словарь. Часть 2. Оптически считываемые носители данных.
3. Bravo-Montero F., Castells-Rufas D., Carrabina J. High-speed laser marking with diode arrays // Optics & Laser Technology. 2022. Vol. 146. Article 107551. DOI: 10.1016/j.optlastec.2021.107551.
4. Ren C., Xu Y., Liu X., Liu H. An automotive tire visual laser marking robot system based on multi-information fusion // Intelligence & Robotics. 2024. Vol. 4, № 4. P. 422-438. DOI: 10.20517/ir.2024.25.
5. Ofoegbu S. U., Rosa P. J. A., Fernandes F. A. O., Pereira A. B., Fonseca P. Laser marking on polyoxymethylene (POM) polymer substrate for a lean manufacturing application // The International Journal of Advanced Manufacturing Technology. 2025. Vol. 136. P. 3863-3882. DOI: 10.1007/s00170-024-13178-7.
6. Горбовец М. А., Славин А. В. Кодированная маркировка образцов для высокотемпературных испытаний // Труды ВИАМ. 2019. № 10. С. 125-132. DOI: 10.18577/2307-6046-2019-0-10-125-132.
7. Microsoft. Dynamic-Link Libraries // Microsoft Learn. Windows documentation.
8. Microsoft. LoadLibraryA function (libloaderapi.h) // Microsoft Learn. Win32 API documentation.
9. Microsoft. GetProcAddress function (libloaderapi.h) // Microsoft Learn. Win32 API documentation.
10. Steen W. M., Mazumder J. Laser Material Processing. 4th ed. London: Springer, 2010.

References

1. GOST R ISO/IEC 16022-2008. Automatic identification. Bar coding. Data Matrix bar code symbology specification.
2. GOST R ISO/IEC 19762-2-2011. Information technology. Automatic identification and data capture techniques. Harmonized vocabulary. Part 2. Optically readable media.
3. Bravo-Montero F., Castells-Rufas D., Carrabina J. High-speed laser marking with diode arrays. *Optics & Laser Technology*, 2022, vol. 146, article 107551. DOI: 10.1016/j.optlastec.2021.107551.
4. Ren C., Xu Y., Liu X., Liu H. An automotive tire visual laser marking robot system based on multi-information fusion. *Intelligence & Robotics*, 2024, vol. 4, no. 4, pp. 422-438. DOI: 10.20517/ir.2024.25.
5. Ofoegbu S. U., Rosa P. J. A., Fernandes F. A. O., Pereira A. B., Fonseca P. Laser marking on polyoxymethylene (POM) polymer substrate for a lean manufacturing application. *The International Journal of Advanced Manufacturing Technology*, 2025, vol. 136, pp. 3863-3882. DOI: 10.1007/s00170-024-13178-7.
6. Gorbovets M. A., Slavin A. V. The coded marking of specimens for high-temperature tests. *Trudy VIAM*, 2019, no. 10, pp. 125-132. DOI: 10.18577/2307-6046-2019-0-10-125-132.
7. Microsoft. Dynamic-Link Libraries. Microsoft Learn. Windows documentation.
8. Microsoft. LoadLibraryA function (libloaderapi.h). Microsoft Learn. Win32 API documentation.
9. Microsoft. GetProcAddress function (libloaderapi.h). Microsoft Learn. Win32 API documentation.
10. Steen W. M., Mazumder J. *Laser Material Processing*. 4th ed. London: Springer, 2010.