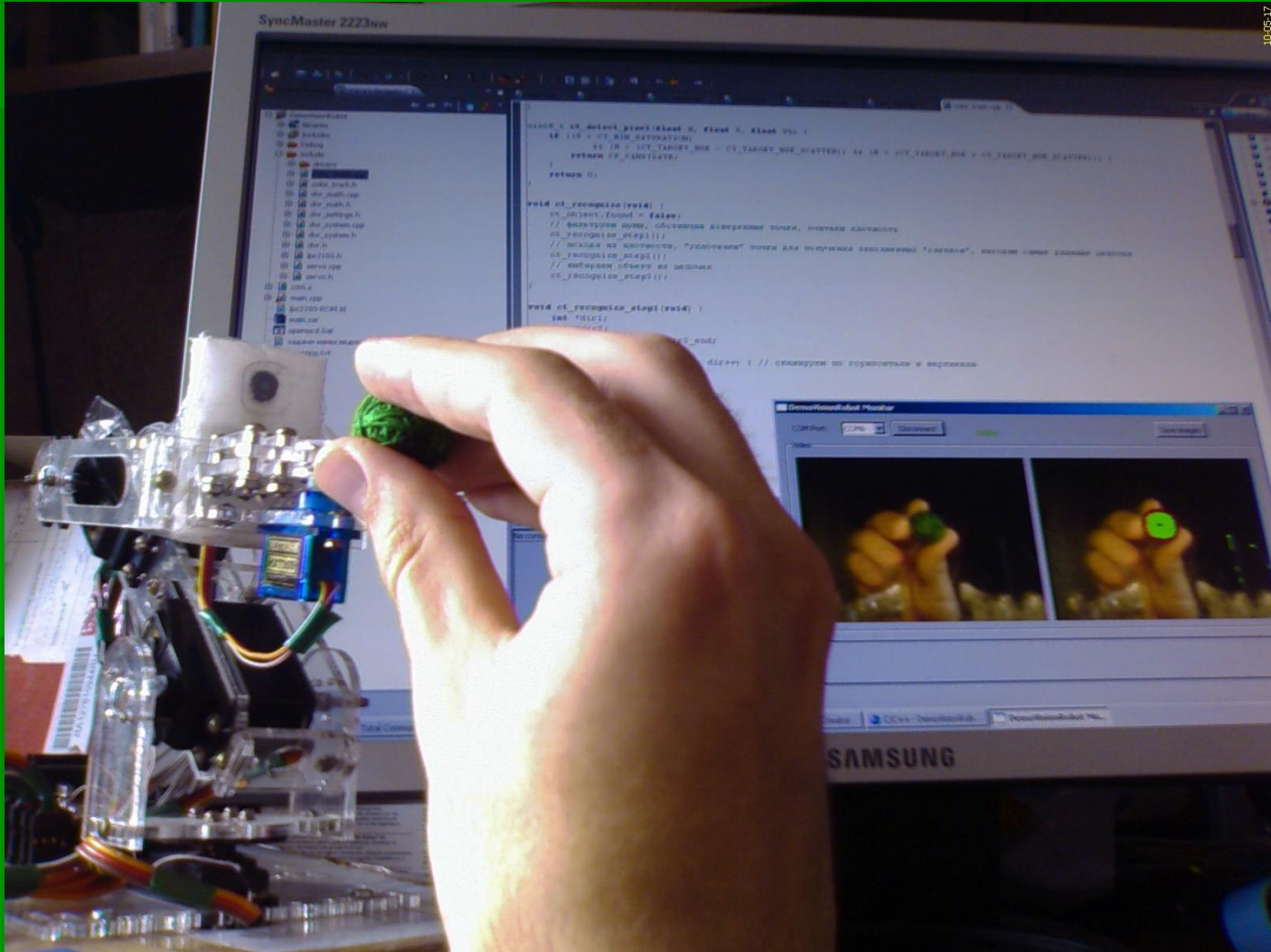


DemoVisionRobot

Робот с техническим зрением, жадный до зеленых шариков



Устройство робота:

- 3-звенный манипулятор *OR-LC-xARM-v1* со схватом *OR-LC-GRIP-HXT900-v1* на последнем звене
- сервопривода HXT12K, HXT5010, HXT900
- модуль *TGA130V10* с видеочкамерой *OmniVision OV9650* (установлен на схвате)
- система управления на базе микроконтроллера *NXP LPC2103* с процессором *ARM7-TDMI-S*

Назначение:

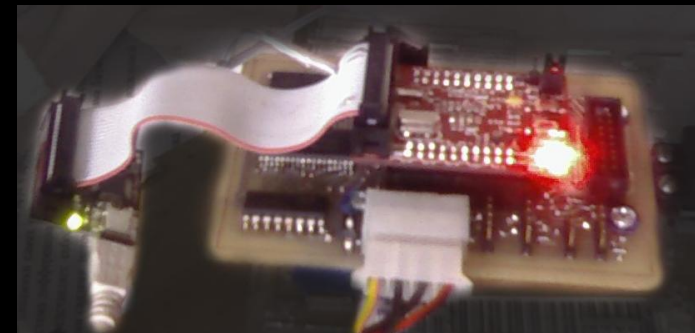
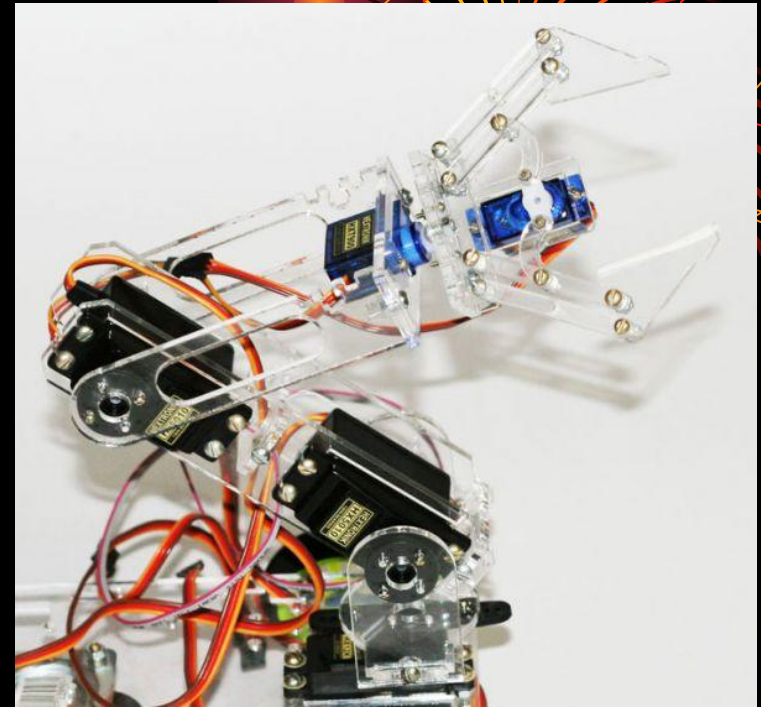
демонстрация технического зрения на примере манипулирования объектами, проведение учебных лабораторных работ

Основной режим работы:

поиск в пространстве зеленого шарика, осуществление его захвата и транспортирование его в корзину

Особенности:

- система технического зрения, позволяющая роботу находить одноцветный сферический объект в пространстве
- полная автономность
- экономически выгодное решение
- сопутствующее программное обеспечение является бесплатным и свободно распространяемым



Математическая модель

Модель описывает кинематику манипулятора. Робот осуществляет **слежение** за объектом, для чего необходимо:

- определять положение схвата в пространстве
- определять положение обнаруженного объекта
- вырабатывать управление

Решение прямой задачи кинематики:

$$X_n = (a_2 \cdot \cos q_2 + a_3 \cdot \cos(q_2 + q_3)) \cdot \cos q_1;$$

$$Y_n = (a_2 \cdot \cos q_2 + a_3 \cdot \cos(q_2 + q_3)) \cdot \sin q_1;$$

$$Z_n = -a_2 \cdot \sin q_2 - a_3 \cdot \sin(q_2 + q_3) + S_1;$$

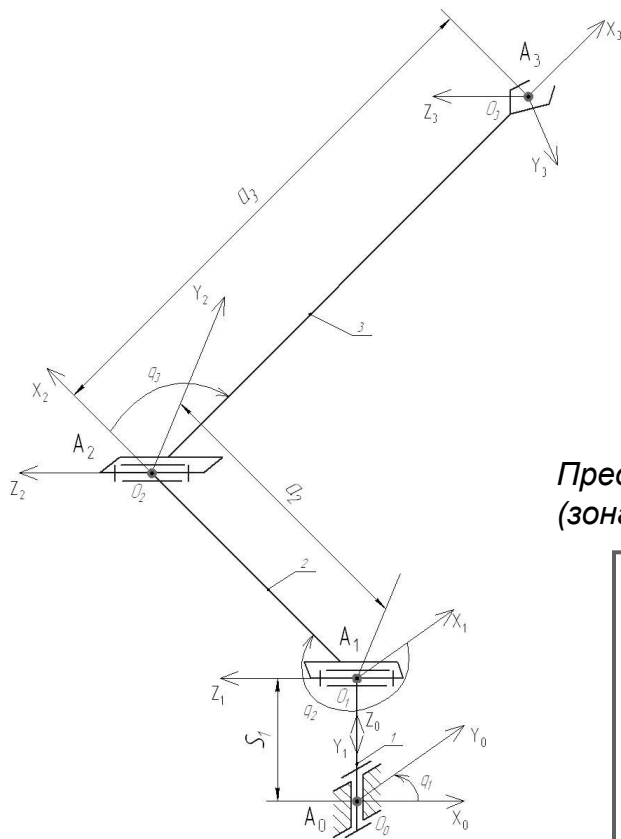
Решение обратной задачи кинематики:

$$Z'_n = Z_n - S_1;$$

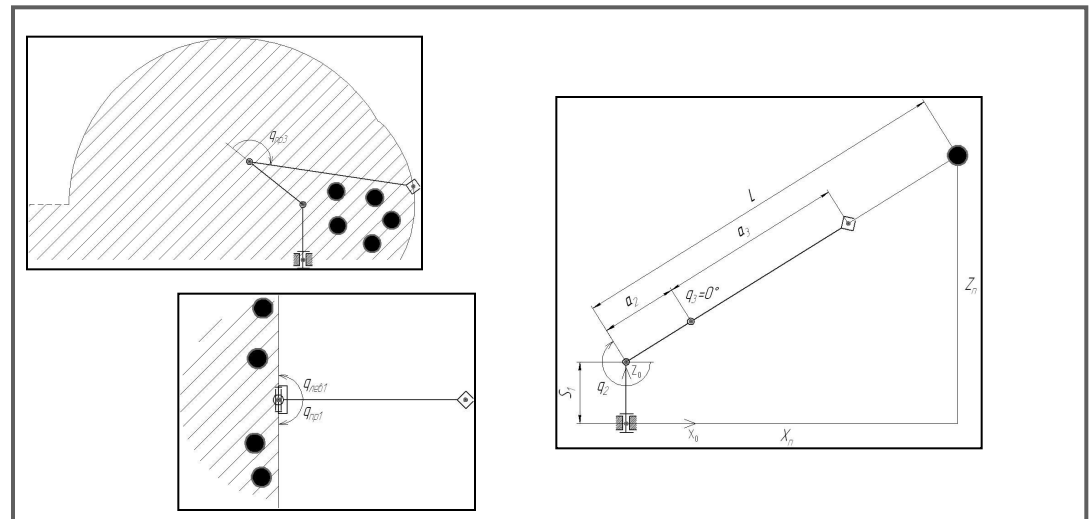
$$q_1 = \arctg \frac{Y_n}{X_n};$$

$$q_3 = \arccos \frac{X_n^2 + Y_n^2 + Z_n'^2 - (a_2^2 + a_3^2)}{2 \cdot a_2 \cdot a_3};$$

$$q_2 = \arccos \frac{a_3 \cdot \sin q_3}{\sqrt{X_n^2 + Y_n^2 + Z_n'^2}} + \arctg \frac{\sqrt{X_n^2 + Y_n^2}}{Z_n'} + 180^\circ;$$

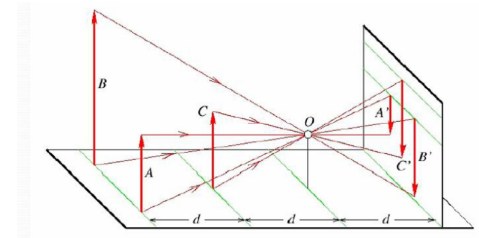
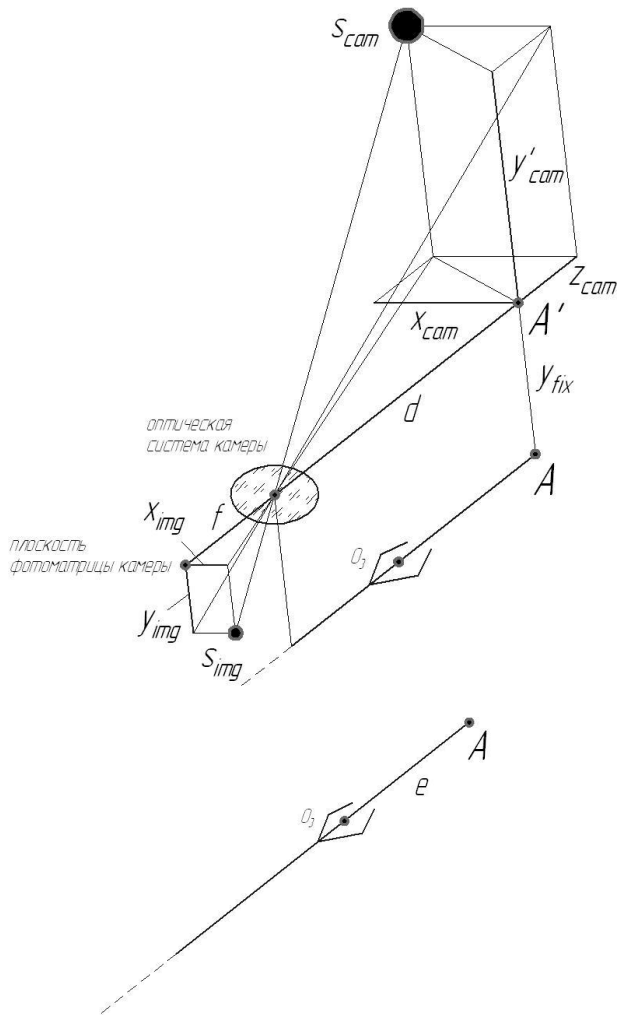


Предусмотрены следующие ситуации (зона недоступности заштрихована):



Математическая модель

Модель оптической геометрии в трехмерном пространстве



Нахождение положения объекта через объектив

$$r_{rel} = \frac{S_{cam}}{S_{img}};$$

$$z_{cam} = r_{rel} \cdot f - d;$$

$$x_{cam} = r_{rel} \cdot x_{img};$$

$$y'_{cam} = r_{rel} \cdot y_{img};$$

$$y_{cam} = y'_{cam} + y_{fix};$$

Нахождение следующего положения схвата при слежении

$$(x_{img} \neq 0, y_{img} \neq 0, z_{img} \neq 0)$$

$$X_n = X_n^* + (z_{cam} \cdot \cos(q_2 + q_3) + y_{cam} \cdot \sin(q_2 + q_3)) \cdot \cos q_1 - x_{cam} \cdot \sin q_1;$$

$$Y_n = Y_n^* + (z_{cam} \cdot \cos(q_2 + q_3) + y_{cam} \cdot \sin(q_2 + q_3)) \cdot \sin q_1 - x_{cam} \cdot \cos q_1;$$

$$Z_n = Z_n^* + z_{cam} \cdot \sin(q_2 + q_3) - y_{cam} \cdot \cos(q_2 + q_3);$$

Нахождение следующего положения схвата при захвате

$$(x_{img} = y_{img} = z_{img} = 0)$$

$$z_{cam} = r_{rel} \cdot f - d + e;$$

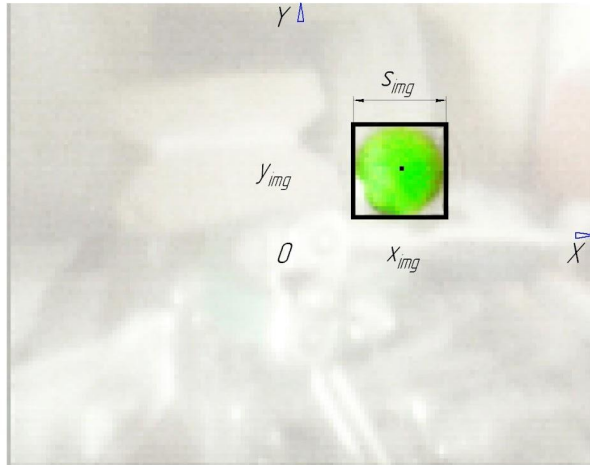
$$X_n = X_n^* + (z_{cam} \cdot \cos(q_2 + q_3)) \cdot \cos q_1;$$

$$Y_n = Y_n^* + (z_{cam} \cdot \cos(q_2 + q_3)) \cdot \sin q_1;$$

$$Z_n = Z_n^* + z_{cam} \cdot \sin(q_2 + q_3);$$

Техническое зрение

Иллюстрация распознавания объекта на сцене перед видеокамерой



Преобразование цветового пространства

$YUV(YC_b C_r) \rightarrow RGB \rightarrow HSV$

H (hue) – тон (цвет)

S (saturation) – насыщенность

V (value) – значение (яркость)

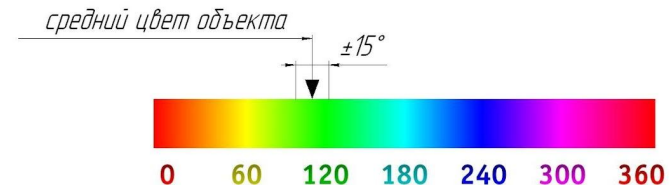
Гистограммы монохроматических уровней (V):
исходная и скорректированная



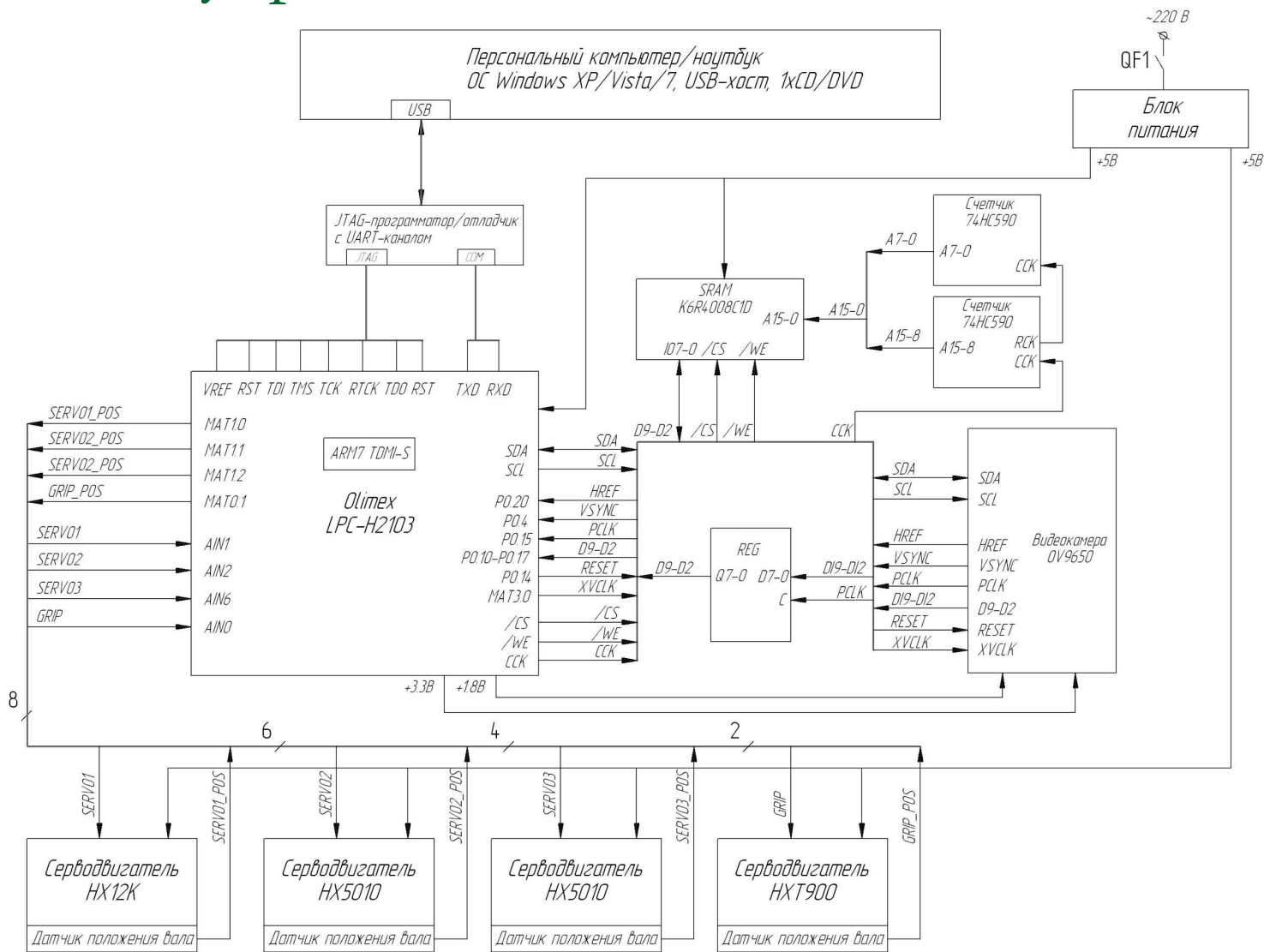
Этапы распознавания:

1. Выделение пикселей на всем изображении, значения цвета и насыщенности которых находятся в искомом диапазоне
2. Фильтрация помех
3. Уплотнение смежных выделенных областей
4. Выбор области, наиболее близкой по форме к окружности
5. Определение габаритных размеров выбранной области

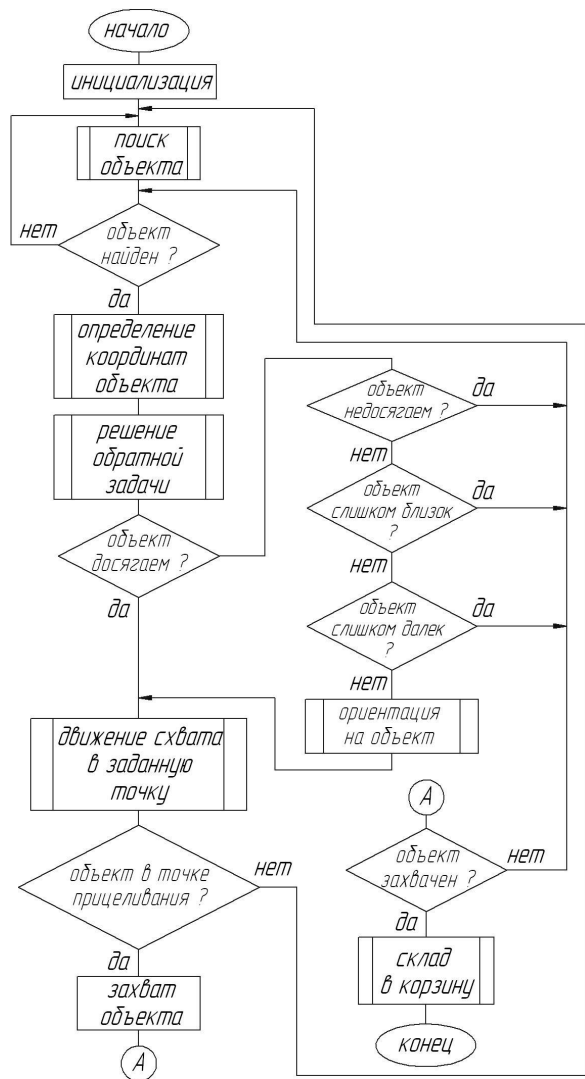
Шкала тона (цвета) – Hue



Система управления



Алгоритмы управления



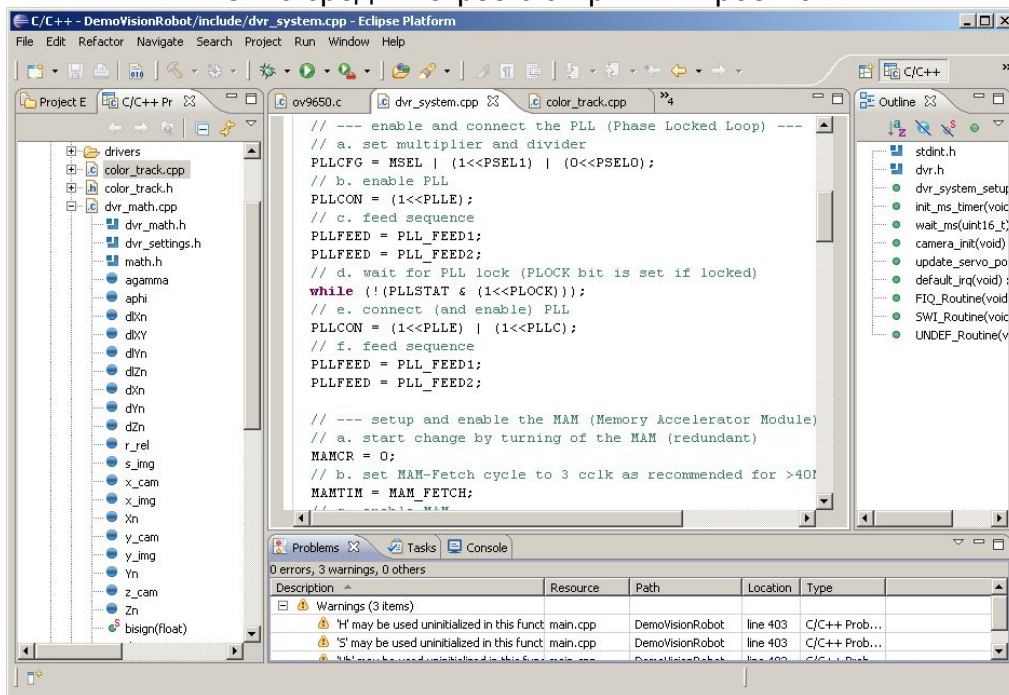
Уровни управления:

- Интеллектуальный
- Тактический
- Исполнительный



Программирование и отладка микроконтроллера

Окно среды Eclipse с открытым проектом



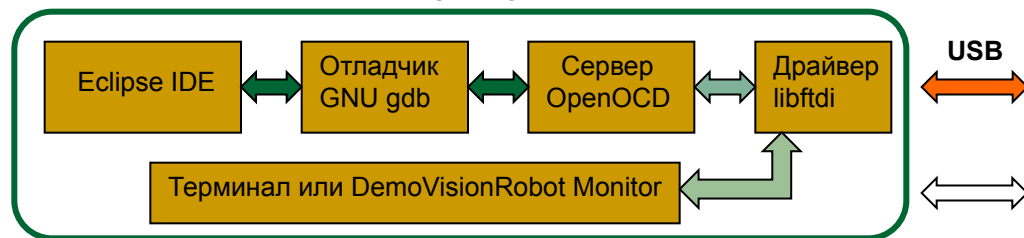
Система управления имеет интерфейсы для перепрограммирования робота:

- JTAG, перепрограммирование + отладка в чипе (OnChip Debugging)
- UART (COM-порт), перепрограммирование

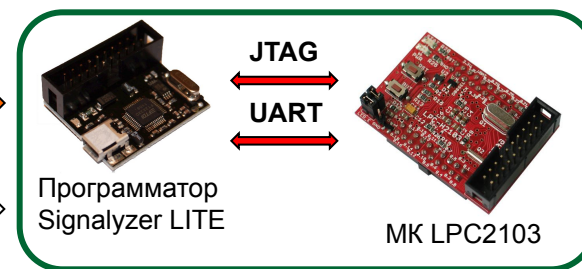
Средства разработки:

- Yagarto GNU ARM Toolchain, пакет средств сборки GNU GCC, библиотека newlib и вспомогательные утилиты
- OpenOCD, средство отладки в чипе
- Eclipse IDE, интегрированная среда разработки на C++

ПК (ноутбук)

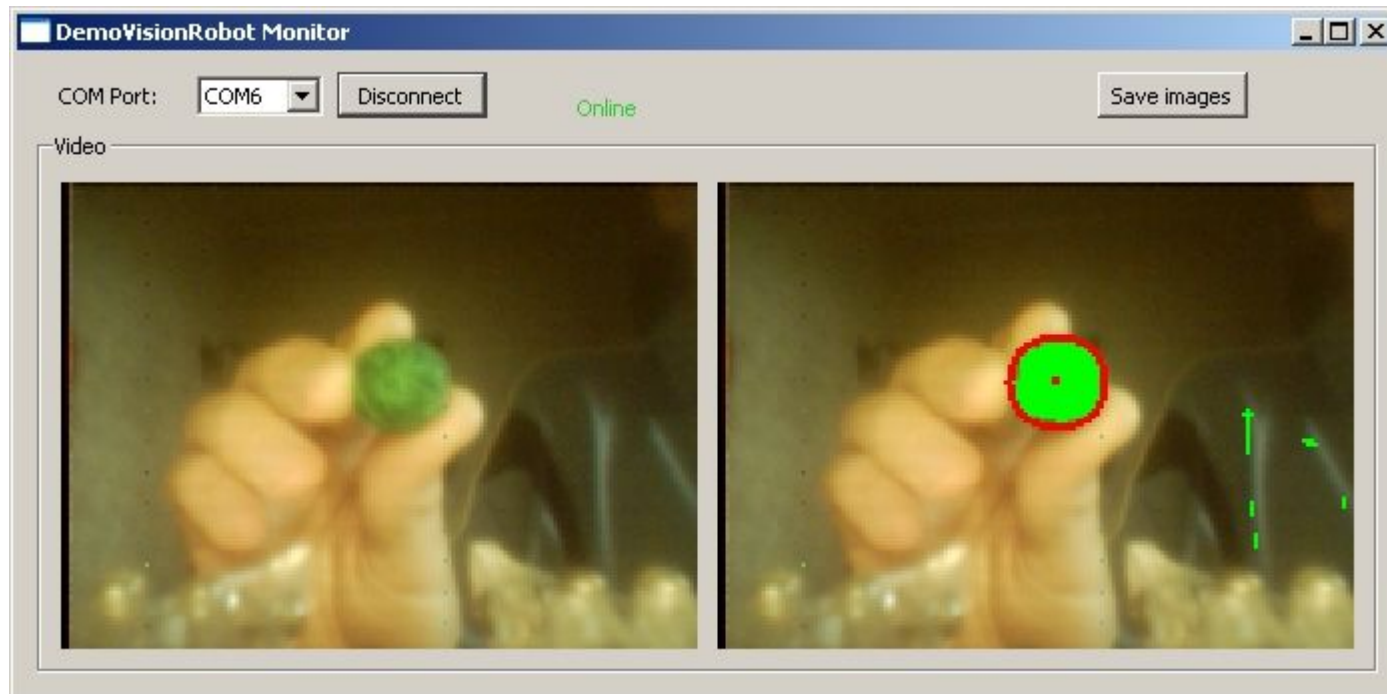


DemoVisionRobot



Программирование и отладка микроконтроллера

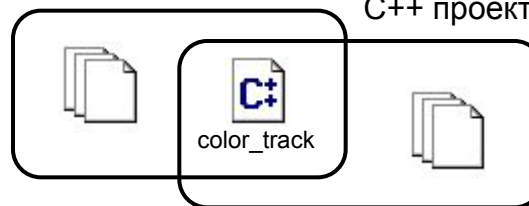
Программа-монитор, написанная на кроссплатформенной библиотеке Qt в среде Qt Creator

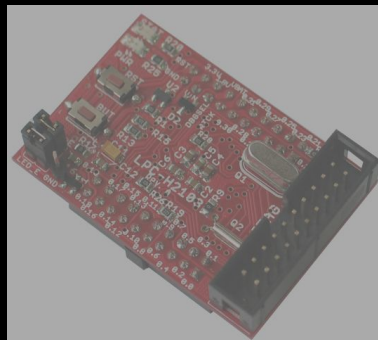


Код алгоритма распознавания объекта содержится в одном файле-модуле, используемом обоими проектами

C++ проект для МК

C++ проект для ПК-хоста





Отработка круговой траектории
(Основная функция еще не реализована)