

# Система с использованием ПИД- регулятора для задачи следования по линии.

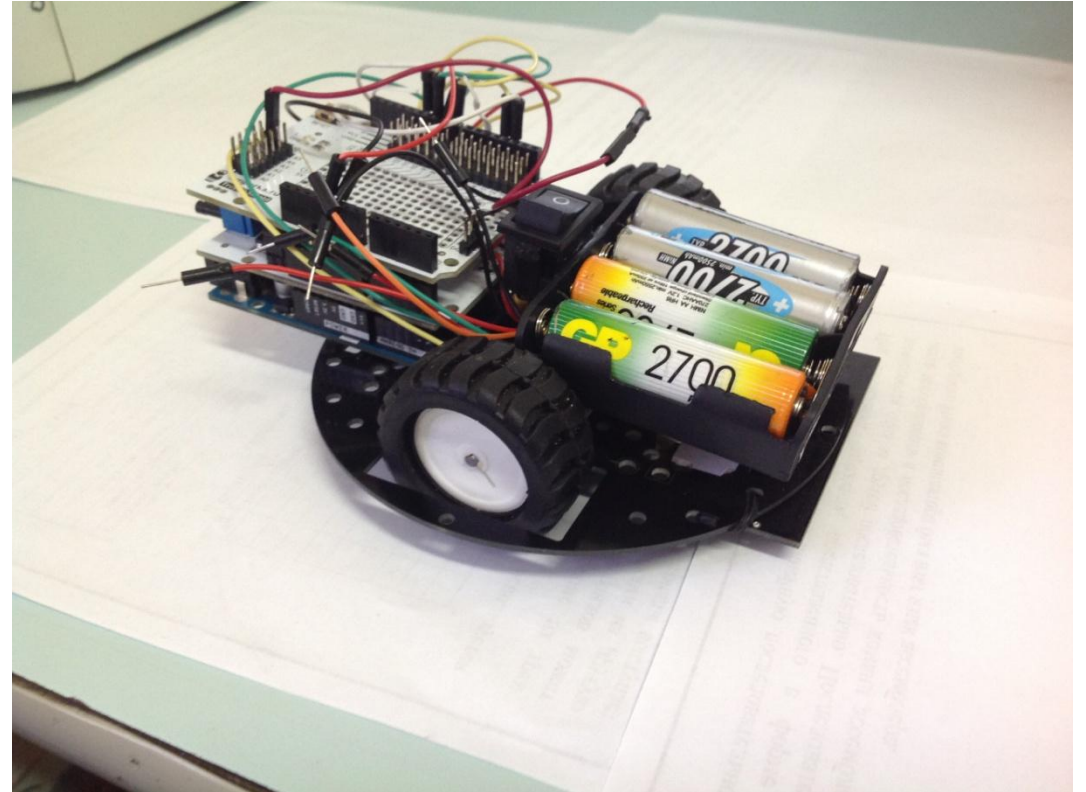


Авторы: Садовин Виктор,  
Татаринов Артём.  
Обучающиеся гр. КС-41, ВК  
ПГТУ «Политехник»  
Научный руководитель: к.т.н.  
доцент кафедры ИВС, Морохин  
Д.В.

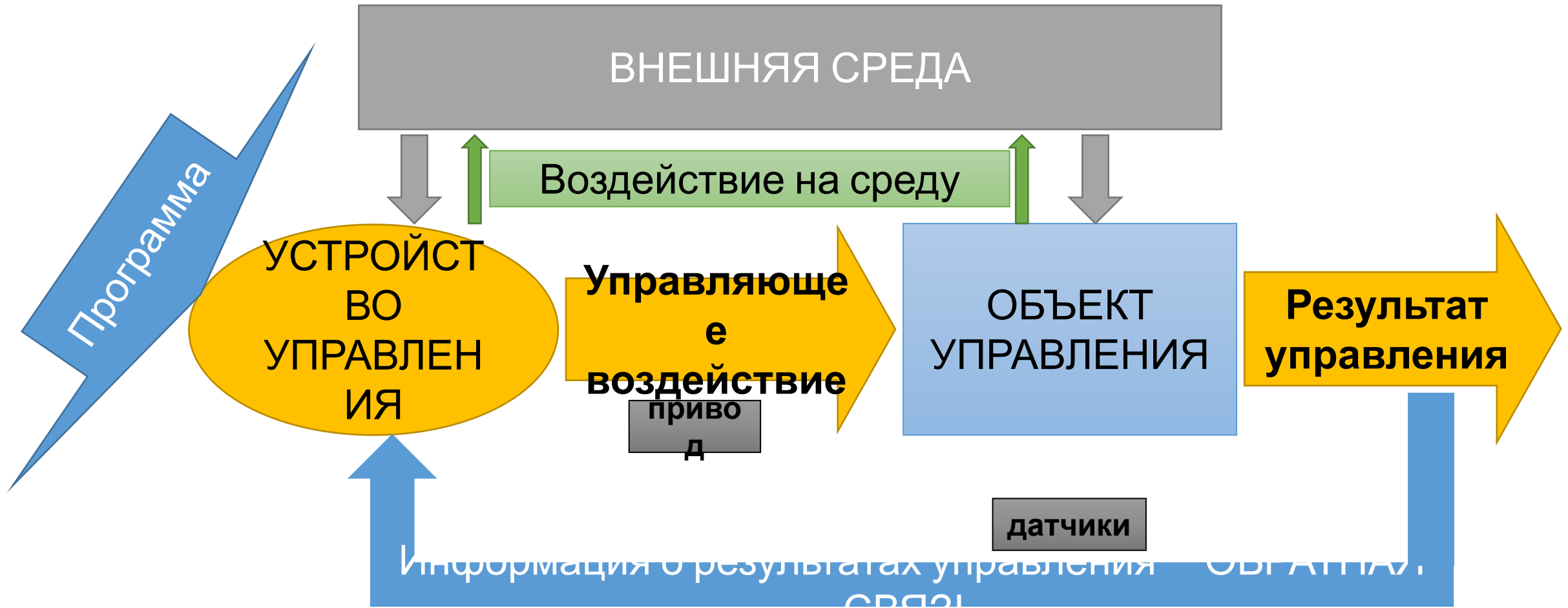
# Робот.

В нашей системе собран робот:

- На двухколёсной платформе
- микроконтроллер Arduino Uno с платой расширения Motor shield и Troyka shield.
- датчик линий QTR Reflectance Sensors фирмы Pololu.



# Основы теории управления.



# ПИД-регулятор.

Формула ПИД-регулятора:

$$u(t) = P + I + D = K_p e(t) + K_i \int_0^T e(t) dt + K_d \frac{de(t)}{dt}$$

Можно заменить на дискретный вариант:

$$u(t) = P(t) + I(t) + D(t);$$

$$P(t) = K_p * e(t);$$

$$I(t) = I(t - 1) + K_i * e(t);$$

$$D(t) = K_d * \{e(t) - e(t - 1)\};$$

# Видео сравнение

# Решение задачи.

```
void line_go(int err) //PID
{
    int proportional = err * 200;
    integral += proportional;
    int derivative = proportional - last_proportional; // Compute the derivative (c
    last_proportional = proportional; // Remember the last position.
    int power_difference = proportional / 10 + derivative * 3 + integral / 10000 ;
    const int max = 120;
    if (power_difference > max) power_difference = max;
    if (power_difference < -max) power_difference = -max;
    if ((power_difference < -30) || (power_difference > 30))
        goForward(max + power_difference, max - power_difference, timer);
    else //power_difference == 0
        goForward(255, 255, timer);
}
```

## Выводы:

- Изучили литературу по ПИД-регулированию.
- Улучшили навыки программирования.
- Получили стабильную и скоростную систему.
- Модернизируем аппаратную часть.
- Улучшим логику работы программы.



**Спасибо за внимание!**