

ВВОД КОМАНД УПРАВЛЕНИЯ С ПОМОЩЬЮ ПУ XBOX360*Назаров А.А., Краси́ло М.С.**Донской государственной технической университет, г.Ростов-на-Дону***Ключевые слова:** робототехника, мехатронный объект, пульт управления, ODROID.**Аннотация.** Цель данной статьи – использовать в качестве ПУ обычный геймпад от xbox360. В статье рассмотрен программный код, отвечающий за обработку полученных команд и объяснены основные моменты и этапы в написании этого участка кода.**ENTER CONTROL COMMANDS USING THE XBOX360***Nazarov A.A., Krasilo M.S.**Don state technical university, Rostov-on-Don***Keywords:** robotics, mechatronic facility, control panel, matrix, ODROID.**Abstract.** The purpose of this article is to use the regular gamepad from xbox360 as a PU. The article discusses the program code responsible for processing the received commands and explains the main points and steps in writing this section of code.

Для управления трехколесной платформой был выбран ПУ от xbox360, с целью удобства в управлении, нежели каким-либо другим пультом.

Помимо управления движением, возникла необходимость в управлении другими компонентами:

1) двигатель управления положением лыж; 2) насосом для захвата предметов; 3) колеса для движения.

Данный робот был спроектирован для соревнований по робототехнике «Эконет», то есть цель этой платформы – привлечь внимание к состоянию окружающей среды и необходимости беречь её.

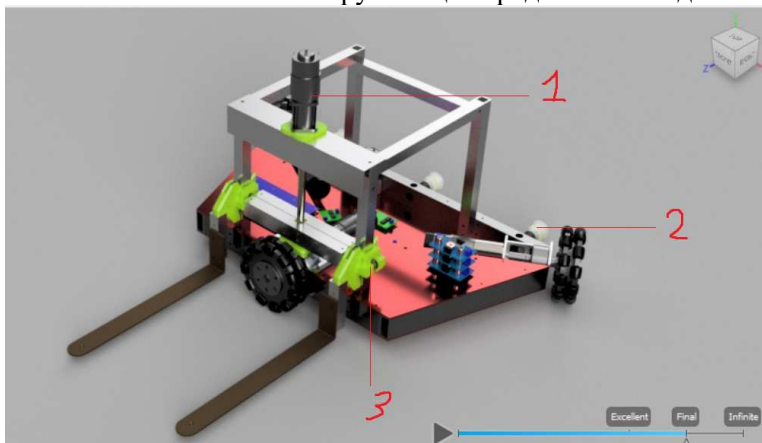


Рис. 1. Обозначение управляемых частей

Для работы был выбран программный продукт EmBitz 1.11 а также микроконтроллер STM32f407. Также был задействован микрокомпьютер ODROID(рисунок 2). Подсоединенный к нему Bluetooth-адаптер позволяет с легкостью получать сигнал от ПУ и передавать его на STM.



Рис. 2. ODROID, внешний вид

Для осуществления управления потребуются два проекта – первый будет вшит в микрокомпьютер, второй в трехколесную платформу.

Рассмотрим проект для микрокомпьютера.

Для определения, на какие кнопки именно нажимает оператор, были написаны функции для каждой кнопки:

```
void Rmove_cmd(float x, float y)
{
```

```

uint8_t packet[PACKET_SIZE];
float args[2] = {x, y};
build_packet(RMOVE_CMD, &args, &packet);
write_packet(&packet);
}

```

Функция Rmove_cmd предназначена для взаимодействия с правым стиком 1 ПУ для движения робота.

```

void action_cmd(int8_t s)
{
    uint8_t packet[PACKET_SIZE];
    int8_t args[8] = {s, 0, 0, 0, 0, 0, 0, 0};
    build_packet(ACTION_CMD, &args, &packet);
    write_packet(&packet);
}

```

Функция action_cmd предназначена для взаимодействия с кнопкой 2 (рисунок 3). При активации кнопки будет включаться насос для захвата предметов.

```

void abort_cmd()
{
    uint8_t packet[PACKET_SIZE];
    uint8_t args[8] = {0, 0, 0, 0, 0, 0, 0, 0};
    build_packet(ABORT_CMD, &args, &packet);
    write_packet(&packet);
}

```

Функция abort_cmd отвечает за изменение программного центра масс робота, что необходимо для расчетов скоростей в дальнейшем. Приводится в действие кнопкой 3 (рисунок 3).

```

void turn_left_cmd(float LTx, float LTy)
{
    uint8_t packet[PACKET_SIZE];
    float args[2] = {LTx, LTy};
    build_packet(TURN_LEFT_CMD, &args, &packet);
    write_packet(&packet);
}

```

Функция turn_left_cmd отвечает за разворот на месте против часовой стрелки, активируется триггером 4 (рисунок 3).

```

void turn_right_cmd(float RTx, float RTy)
{
    uint8_t packet[PACKET_SIZE];
    float args[2] = {RTx, RTy};
    build_packet(TURN_RIGHT_CMD, &args, &packet);
    write_packet(&packet);
}

```

Функция turn_right_cmd отвечает за разворот на месте по часовой стрелке, активируется триггером 5 (рисунок 3).

```

void platform_cmd(int8_t s)
{
    uint8_t packet[PACKET_SIZE];
    uint8_t args[8] = {s, 0, 0, 0, 0, 0, 0, 0};
    build_packet(PLATFORM_CMD, &args, &packet);
    write_packet(&packet);
}

```

Функция platform_cmd отвечает за поднятие/опускание лыж, активируется кнопками 6 и 7 (рисунок 3).



Рис. 3. ПУ и назначенные кнопки.

Для микроконтроллера второй проект состоит из файла Communication.c , в котором происходит обработка полученных с ODROIDa команд:

```
void RightStick_move_command(uint8_t *args)
{
    float x = 0, y = 0;
    memcpy(&x, args, sizeof(x));
    memcpy(&y, &args[sizeof(x)], sizeof(y));
    // применить скорость X, Y
    // TODO: поменять X, Y, на L и R, преобразовывать
    // координаты в напряжение на ODROID'e
    x=(x-0)*(0.45-0)/(32767-0)+0;
    y=(y-0)*(0.45-0)/(32767-0)+0;
    testSpeed[0]=x;
    testSpeed[1]=y;
}
void LeftStick_move_command(uint8_t *args)
{
    float y = 0;
    float x = 0;
    memcpy(&x, args, sizeof(x));
    memcpy(&y, &args[sizeof(x)], sizeof(y));
    //setVoltage(3,y);
    left_stick_X=x;
    left_stick_Y=y;
}
void platform_command(uint8_t *args)
{
    int8_t s;
    memcpy(&s, args, sizeof(s));
    // 1 - поднимать платформу
    // 0 - не двигать платформу
    // -1 - опускать платформу
    if(s == 1)
    {
        setVoltage(3,1);
    }
    else{if(s == -1 {setVoltage(3,-1)} else{setVoltage(3,0)}}
    }
}
void toggle_command(uint8_t *args)
{
    uint8_t a = 0;
    uint8_t b = 0;
```

```

uint8_t x = 0;
uint8_t y = 0;
uint8_t back_but = 0;

memcpy(&a, args, sizeof(a));
memcpy(&b, &args[sizeof(a)], sizeof(b));
memcpy(&x, &args[sizeof(a)+sizeof(b)], sizeof(x));
memcpy(&y, &args[sizeof(a)+sizeof(b)+sizeof(x)], sizeof(y));
memcpy(&back_but, &args[sizeof(a)+sizeof(b)+sizeof(x)+sizeof(y)], sizeof(back_but));
test=a+b+x+y+back_but;
if(back_but !=0)
{
flag = flag *(-1);
}
if(flag == -1)
{
MatrixSmech[0][0] = 1;
MatrixSmech[1][1] = 1;
MatrixSmech[2][2] = 1;
MatrixSmech[1][2] = 0.5;
}
if(flag == 1)
{
MatrixSmech[0][0] = 1;
MatrixSmech[1][1] = 1;
MatrixSmech[2][2] = 1;
MatrixSmech[1][2] = 0.0;
}

```

В этом участке происходит обработка полученных данных, что приводит ко входу в соответствующую функцию и далее исполнительные команды выдают значения, основываясь на полученных ранее.

Таким образом, рассмотрены участки кодов для взаимодействия ПУ и микроконтроллера по беспроводной связи через микрокомпьютер ODROID.

Список литературы

1. Марченко А.Л. М30 Основы электроники. Учебное пособие для вузов. – М.: ДМК Пресс, 2008. – 296с.
2. Подбельский В.В., Фомин С.С. Курс программирования на языке Си: учебник. – М.: ДМК Пресс, 2012. – 384 с.
3. Литвиненко Н.А. Технология программирования на C++. Win32 API-приложения. – СПб.: БХВ-Петербург, 2010. – 288 с.

Сведения об авторах:

Назаров Александр Александрович – студент, ДГТУ, г.Ростов-на-Дону;

Красило Михаил Сергеевич – студент, ДГТУ, г.Ростов-на-Дону.