

***Проект***

***по***

***Операционни Системи***

изготвил: Момчил Петков

ФКСУ 121211019

47 група

Съдържание:

**Bot Board II Ръководство за свързване**

**\*хардуерна информация**

**\*джъмпери по платката**

**\*отстраняване на проблеми**

**Bot Board II Ръководство**

**\*хардуерна информация**

**\*джъмпери по платката**

**\*форматиране на команди**

**\*отстраняване на проблеми**

**Хоби серво**

**Bot Board II**



**Bot Board II Хардуерна информация.**

Снимката по горе показва джъмперите (черните правоъгълници) сложени фабрично при покупка на платката. Джъмперите се използват за операции с мараметри на платката. Не е задължително да са поставени правилно за вашия проект, поради промени които може да правите по логиката и действието което извършва платката. Потърсете в ръководството за проекта ви за правилното поставяне на джъмперите на определените за тях позиции.

Развойната платка Bot Board II има размери 3.00" x 2.30" със 0.125" размер на дупки за поставяне, намиращи се на 0.15" от ъглите.

1. 3 LED диода и бутони се използват само за 3 I/O връзки за изработката на опростен потребителски интерфейс. Правейки I/O връзката със слаб изход LED диод може да бъде включен. С правенето на I/O връзката входна I/O връзка може да се установи дали бутона е натиснат. Не, правете I/O връзката с висок (high) изход и да натискате бутон, понеже може да увредите I/O пиновете. Окъсяват се пиновете оцветени в син цват за ползването на LED-овете и бутоните.

|  |  |
| --- | --- |
| **Чип I/O** | **PB/LED** |
| Пин 12 | A | Червен |
| Пин 13 | B | Зелен |
| Пин 14 | C | жълт |

1. Този порт е за свързване на Sony Playstation джойстик за контролиране на робота. Джъмперите на пинове в синьо се махат, или може да използвате пинове 12-15, които са електрически свързани по същият начин. Забележете, това не работи с Basic Stamp 2, защото изисква инвертер на clock сигнала. Lynxmotion, Sony, и Madcatz контролерите които тествахме изискват 1K завишаване на Пин 12, вече вградено в платката. Други марки може да искат завишаване на Пин 15, което не е включено. Забележете, някои контролери изискват зелената връзка да е свързана с 7.5vdc за активирането на вибромотори или безжично опериране.Playstation 2 оцветяването на кабелите се различава в различните продукции на контролерите. Погледнете таблицата в [следващата](http://www.lynxmotion.com/images/html/build151.htm#aglance) секция за сверяване на цветовете и конекторите за вашия контролер.
2. Регулатора за ниско напрежение Ви дава 5vdc напрежение до 5.5vdc. Изполва се при управление на робота ви с батерии. Може да приема напрежение до 9vdc. Регулатора е направен за стойности до 500mA, но го ограничаваме до 250mA за да предотвратим прегряване.
3. LED за включено състояние. Когато подадете правилно напрежение на регулатора,зелен LED диод се включва.
4. Това е VL вход (логическо напрежение) . Този вход се изполва с 9vdc батерия за да захрани микроконтролера и всичко свързано към 5vdc връзката наплатката. Този вход може да се използва за различаване между VL от VS (серво напрежението).

|  |  |
| --- | --- |
| **Платка** | **Вход** |
| VL + | Червен |
| VL - | черен |

1. Това е VS (серво напрежение) входа. Серво напрежението може да бъде от 4.8vdc до 7.2vdc. Някои микро сервота не позволяват повече от 6vdc. Този вход се използва само за захранване на серовтат, или за захранване на VL и VS използвайки VS=VL джъмпер. Вижте част 7.

|  |  |
| --- | --- |
| **Платка** | **Вход** |
| VS + | Червен |
| VS - | черен |

1. С това захранвате сервотата и логическата част от една батерия. Свързва VS и VL входовете. ВНИМАНИЕ, Използвайки тази опция не свързвайте VL входа.
2. Позволява VL и/или VS входовете да се свържат към 2 Atom 28's аналогови входа с 4:1 напрежително разделяне. Например, ако напрежение на батерията е 9vdc, аналоговия вход ще е 2.25vdc.

|  |  |  |
| --- | --- | --- |
| **Чип I/O** | **Atom Pro ПИН** | **Цел** |
| AX0 | P16 | V-Servo / Основна употрба |
| AX1 | P17 | V-Logic / Основна употрба |
| AX2 | P18 | Основна употрба |
| AX3 | P19 | Основна употрба |

1. Има опция за говорител. За използвантето му, изпраща се правилна команда за генериране на звук към Пин 9. Забележете, че I/O пин се управлява от усилвател. Не се свтрзва директно с говорителя.
2. Включете M/F DB9 кабел от отзи вход за 9-пинов сериен порт към компютър за програмиране на процесора и получаване на дебъгваща инфорация. Алтернативен USB-към сериен адаптер става също. Вижте частта с [USB](http://www.lynxmotion.com/images/html/build151.htm#usb2s) към сериен адаптер за повече информация.
3. Когато натиснете този бутон,ще нулирате микро процесора. Това може да бъде полезно за стартиране на различни програми,в зависимост от това кой SUI бутон е натиснат при рестартирането.
4. Тук се включва хибридния модул. Този модул може да е Atom 28, Atom Pro 28, Basic Stamp 2 серии, OOPic-C, BASICX 24, или всеки BS2 пиново съвместим. Atom модулите пресоставят допълнителни 4 I/O пина.
5. Тук се конфигурира I/O шината за използване на VL (+5vdc от платката) или VS (директно от захранващият блок на серво входа). Разделени са на сектори по 4 I/O пина. Внимание, свързване на напрежението от сервото към тези блокове с 5vdc периферия, ще повреди периферията.
6. Тук се свързват сервотата, двигателни контролери, сензори, и др. Към микроконтролера. Внимавайте при свързване към I/O шината. Не свързвайте каквото и да е при включено напрежение

**ДЖЪМПЕРИ И КОНЕКТОРИ.**

.

****

Отстраняване на проблеми

Зеленият LED за захранването трябва да свети ако е подсигурено подходящото напрежение. Бъдете сигурни че свързвате батерията с правилния поляритет. Червеното е (+), а черното маса (-). При обратно свързване компонентите по платката могат да бъдат константно повредени! Проверете окабеляването внимателно преди подаване на напрежение.

Максималното напрежение за VL е 9.6vdc. Ако използвате по високо напрежение регулатора може да прегрее и да се изгорите!

Позиционирайте микроконтролера с ПИН 1 към стрелката на рестартиращият бутон. Ако е наобратно няма да бъде повреден от Bot Board, но няма да бъде разпознат от IDE средата.

**Програмиране на - USB към серийни кабели**

Ако сте нов с това проверете: [programming tutorial](http://www.lynxmotion.com/images/html/build147.htm).

Най-бързият начин за програмиране на Atom или Atom Pro е с PC с реален сериен порт. Ако имате настолно PC без сериен порт се препоръчва да сложите външна такава карта. Това е [мини](http://www.lynxmotion.com/images/html/build126.htm) ръководство за добавяне на сериен порт. Ако използвате лаптоп или PC без сериен порт може да ползвате USB към сериен кабел.

Забележка: FTDI USB кабелите имат настройка към драйверите им наречена Latency и Buffer. Намалете буфера от 4k на 1k, а Latency от 16 на 1.

**SSC-32.**



**SSC-32 Хардуерна информация.**

Снимката по горе показва джъмперите (черните правоъгълници) поставени на платката както е фабрично зададена. Джъмперите се използват за поставяне на параметри за платката. Предполага се че няма да са на правилното място в зависимост от нуждите на вашия проект. Please consult the tutorial for your project for the proper shorting bar jumper positions.

SSC-32 има размери 3.00" x 2.30" с 0.125" дупки за закрепване на 0.15" от ъглите на платката.

1. Трансформатор за регулиране на напрежението ви предоставя 5vdc източник с 5.5vdc посотянно напрежение. Това е нужно при работа на робота с батерия. Може да приеме максимално 9vdc inвходно напрежение. Трансформатора е изчислен за 500mA, но го използваме до 250mA за да не се стигне до прегряване.
2. От тук се захранват сервота от 16 до 31. Приложете от 4.8vdc до 6.0vdc за повечето видове сервота. Това може да се направи директно от 5-клетъчна NiMH батерия. 7.2vdc - 7.4vdc може да се приложи на HSR-5980 или HSR-5990 сервота. Може да ползвате 6-клетъчнаl NiMH или 2-клетъчна LiPo батерии.

|  |  |
| --- | --- |
| **Платка** | **Вход** |
| VS2 + | Червен |
| VS2 - | Черен |

1. Тези джъмпери се използват за свързване на VS1 към VS2. Използвайте тази опция когато захранвате всички сервота от една и съща батерия. Монтирайте и двата джъмпера. Алтернативно, ако искате да ползвате отделни батерии, за всяка страна, премахнете джъмперите.
2. Това е логическото напрежение, или VL. Този вход нормално се ползва с 9vdc батерия и захранва ICs и всичко свтрзано към 5vdc връзката на платката. Стойностите за този отсег са между 6vdc - 9vdc. Този вход за използва за разграничаване на логическата връзка от захранването на сервото. Нужно е да се премахне VS1=VL джъмпера при захранване на сервота отделно от VL. SSC-32 използва 35mA без да е свързано друго към 5vdc изхода.

|  |  |
| --- | --- |
| **Платка** | **Вход** |
| VL + | Червен |
| VL - | Черен |

1. Този джъмпер позволява захранване на микроконтролера и поддържа връзка за захранване на сервото. Нужни са поне 6vdc за да работи правилно. Ако контролера се нулира при движение на прекалено много сервота в един момент, може да се наложи да захраните микроконтролера отделно от VL входа. 9vdc е добро напрежение за това. Този джъмпер трябва да се премахне при самостоятелно захранване на сервотата!
2. Тази връзка захранва серво канали от 16 до 31. . Приложете от 4.8vdc до 6.0vdc за повечето видове сервота. Това може да стане от 5-клетъчна NiMH батерия. 7.2vdc - 7.4vdc може да се приложи на HSR-5980 или HSR-5990 сервота. Може да ползвате 6-клетъчнаl NiMH или 2-клетъчна LiPo батерии.

|  |  |
| --- | --- |
| **Платка** | **Вход** |
| VS1 + | Червен |
| VS1 - | Черен |

1. Тук свързвате сервота или други изходни устройства. Внимавайте и спирайте захранванията при промени по I/O шината.

За конкретни изходи (Hi / Low), всеки изход може да се синхронизира или да подава на 35mA. Все пак има максимален ток от 70mA за 8 I/O групи от пина : 0-7, 8-15, 16-23, 24-31.

|  |  |
| --- | --- |
| **Платка** | **кабел** |
| Пулс | Жълт или бял |
| VS | червен |
| Ground | Черен или кафяв |

1. Тук се слага Atmel IC чипа. Внимавайте да го сложите с Пин 1 към горният десен ъгъл както е показано. Не огъвайте пиновете.
2. двата BAUD входа ви позволяват настройка на baud стойност. Погледнете примерите по - доло.

|  |  |  |
| --- | --- | --- |
| **джъмпери** | **Baud Rate** | **използване** |
| 0 0 | 2400 | Забавя процесора |
| 0 1 | 9600 | Забавя процесора |
| 1 0 | 38.4k | Atom/Stamp свързване |
| 1 1 | 115.2k | PC свързване, Firmware обновяване |

1. ABCD входовете имат и двете поддръжки – статични и latching. Входовете имат вътрешни ниски (50k) pullups които се използват при команда Read Digital Input. Нормалнотоворен ключ свързан от изхода към масата върши работа. Има 2 пина с по 5vdc, два с маса, и 4 с надпис A, B, C, и D.
2. Това е LEDза състоянието на процесора(ако е добро). Свети посотянно при наличие на мощност и остава така докато процесора получава информация от серийния порт. Изгасва и мига след това при получаване на отделна серийна информация. Забележете, че LED изгасва дори ако информацията е с грешна baud стойност, форматирана грешно, или тотално объркана. Полезно е да знаете че SSC-32 получава информация или не.
3. Включете прав M/F DB9 кабел към свободната 9-пинова серийнавръзка на вашия PC за регулиране на серзо позициите. Алтрнативно и USB-към-сериен кабел върши работа.
4. Това е 8-пинов EEPROM сокет. EEPROM е поддържан от 2.01GP firmware. Поддържа 24LC256 който е 32KB чип. 24LC512 чип памет работи но по голям запис на информация не би могъл да поеме.
5. Това е TTL сериен порт или DB9 такъв. Сложете два джъмпера както е показано за да отключите DB9 порта. Включете свръзка за TTL серийна комуникация от микроконтролер.

**Джъмпери и конектори на платката**

****

**Форматиране на команди за SSC-32.**

**Типове команди и групиране.**

Изключвайки MiniSSC-II режима, всички SSC-32 команди свършват със знак номер 13 от ASCII. Няколко команди от един и същи тип могат да бъдат изпълнени заедно в ***командна група***. Всички комадни в тази група ще се изпълнят след престигането на знака за край на всичките взети заедно. Команди от различни типове не могат да се комбиниратпо групи. Числовите команди към SSC-32 трябва да са ASCII низове от десетични числа, например "1234". Може да има команда с отрицателни числа: "-5678". ASCII формата не е чувствителен на главни букви. Изплзвайте колкото байта са нужни. Интервали, табулации, и нов ред се игнорират.

|  |
| --- |
| **Типове команди и групи** |
| 1 | Движение на серво | 7 | Чете аналогов вход |
| 2 | Дискретен изход | 8 | 12 Servo Hexapod Gait Sequencer |
| 3 | Байтов изход | 9 | Query Hex Sequencer |
| 4 | Опашка на движенията | 10 | Дава версията |
| 5 | Опашка размер на пулсове | 11 | Отива на Boot |
| 6 | Чете цифров вход | 12 | MiniSSC-II съвместимост |

**Серво или групово движене.**

|  |
| --- |
| **# <ch> P <pw> S <spd> ... # <ch> P <pw> S <spd> T <time> <cr>** |
| <ch> | Номер на канал десетично, 0-31 |
| <pw> | Дължина на пулса в секунди, 500-2500 |
| <spd> | Скорост на местене в uS за секунда на канал(опционално) |
| <time> | Време в mS за целия ход, за всички канали, 65535 max (опционално) |
| <cr> | дава знак от ASCII 13 (нужен за начало на действие) |
| <esc> | Отмяна на действието, ASCII 27 |

Пример за движение на сервото: "#5 P1600 S750 <cr>"

Примера ще премести сервото на канал 5 на позиция 1600. Ще се премести със стойност 750uS до достигане на позицията. Около 1000uS се равнява на 90° завъртане. Стойност от 100uS значи ч на сервото му трябват 10 секунди за завъртане на 90°.

Пример за движение на сервото:: "#5 P1600 T1000 <cr>"

Примера ще премести сервото на канал 5 на позиция 1600. отнема 1 секунда за хода без значение от каде идва сервото

Серво групов ход: "#5 P1600 #10 P750 T2500 <cr>"

Примера ще премести сервото на канал 5 на позиция 1600 и серво 10 на позиция 750. Отнема 2.5 секунди. Сервотата тръгват и спират да се движат едновременно. Много мощна команда. Командвайки всички крака в режим група лесно се синхронизират комплексни движения.

Може д комбинирате командите за скорост и движение. Скоростта за всяко серво се изчислява по следните правила:

1. Сервотата тръгват и спират да се движат едновременно
2. Ако е зададена скорост за сервото нама да се мести по бързо от зададената скорост, но може да се движи по бавно ако командата за време изисква това
3. Ако е зададено време за хода то хода ще отнеме поне времето което е зададено, но може и повече ако скоростта го ограничава

Серво ход : "#5 P1600 #17 P750 S500 #2 P2250 T2000 <cr>"

В примера се слага1600uS на ch5, 750uS на ch17, и 2250uS на ch2. Хода отнема поне 2 сек.,но ch17 няма да се движи с повече от 500uS. Реалното време ща зависи от дължината на пулса за ch17. Предполагаме, че ch17 започва на позиция 2000. Трябва да мине 1250uS. Ограничен е на 500uS трябват 2.5 сек, така, че хода отнема 2.5 сек. От друга страна, ако ch17 започне на 1000, се мести само с 250uS,прави го за 0.5 сек, хода отнема 2 сек.

ВАЖНО: първата команда трябва да е "# <ch> P <pw>". Понеже контролера не знае каде се е намирало сервото преди това и ще се игнорират други команди

**Софтурено отклонение на позицията.**

|  |
| --- |
| **# <ch> PO <offset value> ... # <ch> PO <offset value> <cr>** |
| <ch> | Канал десетично, 0-31 |
| <offset value> | 100 до -100 в uSeconds |
| <cr> | Знак от ASCII 13 |

Тази команда позволява центриране (1500uS) на сервотата да е точно. Коригира до 15° от позицията. Командата се ползва еднократно в програмата ви. При изключване на SSC-32 забравя тази команда.

**Основна изходна информация.**

Изходите на SSC-32 идват от 4 74HC595 8 битови рег. чипа. Има 4 сектора по 8 bit изходи разделени на: 0-7, 8-15, 16-23 и 24-32. The outputs can sink or source up to 25mA per pin, but a max of 70mA per bank must be observed.

**Дискретен изход.**

|  |
| --- |
| **# <ch> <lvl> ... # <ch> <lvl> <cr>** |
| <ch> | Канал десетично, 0-31 |
| <lvl> | Логическо ниво на канала, 'H' за High или 'L' за Low |
| <cr> | Знак от ASCII 13 |

Канала застава на това ниво до 20mS от задаване на командата.

пример: "#3H #4L <cr>"

изкарва High (+5v) на канал 3 и Low (0v) на канал 4.

**Двуичен изход.**

|  |
| --- |
| **# <bank> : <value> <cr>** |
| <bank> | (0 = Pins 0-7, 1 = Pins 8-15, 2 = Pins 16-23, 3 = Pins 24-31) |
| <value> | Десетично число за bank (0-255), Bit 0 = LSB of bank |
| <cr> | Знак от ASCII 13 |

Тази команда дава 8 bit да бъде записана едновременно. Всички пинове на клетката се обновяват едновременно. Клетките се обновяват до 20mS от подаване на командата.

пример: "#3:123 <cr>"

ще сложи стойността 123 (десетично)в клетка 3. 123 (dec) = 01111011 (bin),и летка 3 е с пинове 24-31. Тази команда ще вкара "0" на пинове 26 до 31, и "1" на останалите.

**заявка на движениета.**

пример: "Q <cr>"

връща "." ако е завършен кода, или "+" ако се изпълява.

Има изчакване от 50uS до 5mS преди пращане на отговор.

**заявка с дължини на пулса.**

пример: "QP <arg> <cr>"

връща байт показваш дължината на пулса на сервото в продължение на 10uS. Например ако дължината е 1500uS, върнатата стойност е 150 (бинарно).

Няколко сервота могат да бъдат зададени в една команда. Връща по байт за серво. Има изчакване от 50uS до 5mS преди отговор обиновенно отговора се получава до 100uS.

**Четене на цифров вход.**

пример: "A B C D AL BL CL DL <cr> "

A, B, C, и D са нормални входове за четене. Четат стойност на входа бинарно. връща ASCII "0" при нисък вход (0v) или ASCII "1" при висок (+5v).

AL, BL, CL, и DL са latching входове. Връщат стойността на входа в ASCII "0" за low (0v) или ако е билlow от последната \*L команда. Връща high (+5v) ако входа е high и не е бил low от последната \*L команда.

 ABCD входовете имат слаб pullup (~50k) при използването им като входове. Проверяват се на 1mS, and are debounced for approximately 15mS.Логическата стойност няма да се смени преди нова логическа стойност да бъде на входа в продължение на 15mS. Тези команди могат да бъдат групирани в едно четене, до 8 стойности наведнъж. Връщат низ с един знак за всеки вход без интервали..

пример: "A B C DL <cr>"

**четене на аналогови входове.**

пример: "VA VB VC VD <cr>"

VA, VB, VC, и VD четат стойността на входа аналогово.връща байт с 8-bit стойност за напрежението на пина.

Когато ABCD се ползват аналогово вътрешния pullup е изключен. Входовете се филтрират цифрово за намаляване на шум. Филтрираните стойности се пращат финализирани до 8mS от промяната. Връщане на 0 показва 0vdc. стойност 255 се отнася за +4.98vdc. за превръщане на върнатата стойност в напрежение, умножете по 5/256. При включване ABCD входовете са настроени за цифров вход с pullup. Първият път на V\* команда, пина ще се преобразува в аналогов без pullup. Няма да върне информация първото четене.

пример: "VA VB <cr>"

този пример връща 2 байта с аналоговите стойности на A и B. Например ако напрежението на Pin A е 2vdc и Pin B е 3.5vdc, ще върне байтове102 и 179.

**Откриване и разрешаване на проблеми**

**Тестване на контролера**

 Най-лесният начин за тестване на контролера е да използвате LynxTerm, нашата безплатна терминална програма. LynxTerm може да свалите от  [тук](http://www.lynxmotion.com/showproduct.aspx?productID=567&CategoryID=15).

Веднъж вече инсталирана, натиснете на Port падащото меню и изберете компорта който ще ползвате. Работи и с USB-към сериен порт адаптери. Сложете джъмперите за 115.2k baud и два DB9 серийни портови джъмпери. включете DB9 M/F кабел от PC към контролера.

Инсталирайте 2 сервота, едно на канал 0 и другото на канал 1.

Включете SSC-32 и вижде зеления LED дали свети.

Изберете терминалния прозорец за да въведете следното, като <cr> значи да натиснете enter.
#0 P1500 #1 P1500 <cr>

И джете сервота заемат централна позиция. LED вече не свети. Ще свети само когато контролера получава данни. Напишете :
#0 P750 #1 P1000 T3000 <cr>

Двете сервота се движат по часовниковата стрелка, серво 0 се движи бавно а серво 1 по бързо. Пристигат на мястото си по едно и също време макар да се вдижат с различни скорости.

#0 P750 <cr>

Това ще накара сервото да се движи в целия си диапазон за 10 сек.
#0 P2250 T10000 <cr>

Докато се движи напишете:
Q <cr>

Контролера трябва да върне "+". Ще върне "." когато сервото спре.

За проба на скоростта:
#0 P750 S1000 <cr>

Движи сервото от 2250 на 750 (към 170°) за 1.5 сек.
изминато разстояние / скорост = време за изминаване
(2250uS-750uS) / (1000uS/Sec.) = 1.5 Sec.

#0 P2250 S750 <cr>

Мести сервото от 750 на 2250 за 2.0 сек.
(2250uS-750uS) / (750uS/Sec.) = 2.0 Sec.

Скорости над 3500 ща местят сервото с максимална скорост не по бързо.

**Отстраняване на основни грешки/проблеми**

Ако видите серво да спре или да не задържа позицията си при движение на няколко сервота, то SSC-32 се е рестартирала. Вижте дали зеления LED свети след инструкция за движение. LED не е индикатор за захрнване, а за състояние. Когато SSC-32 е включена, LED свети постоянно. Свети до получаване на символ и след това мига само при получаване на информация.

SSC-32 има два входа за захранване. VL захранва микроконтролера и неговата поддържаща верига със 5vdc. VS захранва сервотата диектно. При захранване от един източник джъмперите VS1=VL взимат 5vdc от VS буксата. Става за работа на батерия стига напрежението да не падне драстично. Ако напрежението падне то SSC-32 се занулява и рестартира. За да оправите това VS1=VL махате сжъмпера и слагате 9vdc батерия за VL входа.

Единственото друго нещо което може да причини това е лоша захранваща система. Ако окабеляването не е оразмерено, или конекторите не са с добър контакт или се ползва евтина батерия(некачествена). 99% от проблемите с SSC-32 са свързани със захранването. Ако забележите проблеми с движението на сервотата проверете напрежението на захранването им.

.