Teensy BalanceBot Mk1

Курсов проект по
Вградени и автономни
системи

Никола Тотев

Софтуерно Инженерство
3 Курс

ФН: 62271

Въведение

Документацията, която следва предоставя на читателя подробна информация за проекта „Teensy BalanceBot Mk1”

Commented [NT1]: The following documentation will give the reader detailed information about the Teensy BalanceBot Mk1 project.
The goal of this project is to develop a balancing robot from the ground up. This means going through the full process of creating a product/project, everything from System analysis and development of the system to conducting tests to verify that all the objectives are met and a complete and detailed documentation of all the steps.

Целта на проекта е да разработи балансиращ се робот от нулата. Това означава да се мине през пълния процес на създаване на продукт/проект. Всичко от анализ на системата и разработка й, до провеждане на тестове за проверка на функционалност и създаване на подробна документация на всички стъпки.

Анализ на системата

Тази секция покрива желаната функционалност на системата, как може да се моделира робота, какви решение съществуват и как те се сравняват с проекта, който се разработва.

Желана функционалност

Общ преглед

Това е сравнително малък проект, който разполага с ограничено време за разработка и тестване, затова желаната функционалност е ограничена. Характеристиките, които са избрани са достатъчни за създаването на MVP(Minimum Viable Product) - МЖП (Минимално Жизнеспособен Продукт), който може да служи за добра начална точка за бъдещи проект, които са базирани на балансиращ се робот. Бъдещи възможности за развитие и подобрения се обсъждат по-късно в документацията.

Възможни приложения

Въпреки малкия размер на Teensy BalanceBot Mk1, той има няколко възможни приложения

- Първото приложение е в образованието. Робота достатъчно прост, за да може да служи като въведение в роботиката за ученици и студенти, но има много възможности за бъдещо развитие.
- Второто приложение е в изследователската дейност. Роботите стават все по-обичайна гледка в ежедневието ни и различни ситуации изискват роботи с различни характеристики. BalanceBot Mk1 предоставя добра платформа върху която изследователите могат да надграждат.

Функционалност

Commented [NT2]: This section covers desired functionality of the system, how it can be modelled, what solutions currently exist and how they compare to the project being developed.

Commented [NT3]: This is a relatively small project with limited time for development and testing, therefore the desired functionality has been reduced to the bare minimum. The features that have been chosen are enough to create an MVP (Minimum Viable Product) that can be used as a good starting point for future projects that require a balancing robot platform. Future development possibilities and improvements are discussed later in the documentation.

Commented [NT4]: Despite the small formfactor of the Teensy BalanceBot Mk1, it has a couple of use cases.
•The first use case is in education. It is simple enough to be an introduction to robotics for students, but it has enough complexity for any future development.
•The second use case is in research. Robots are becoming very common in everyday life, and different situations require robots with different sets of features and form factors. With the limited feature set, BalanceBot Mk1 provides an excellent blank canvas for researchers to build upon.

Commented [NT5]: **•Контрол на ъгъла** - Angle control is the single most important feature for Teensy BalanceBot Mk1. Angle control goes well beyond staying in an upright position. Consider the situation where the robot is required to move from point A to point B. When repositioning, due to the nature of the robot, it must either lean forward or backward. This requires more consideration when developing the software, because the lean angles when moving are not constant, they depend on the speed of the robot.

•Задържане на позиция -This feature is not critical to the operation of the robot, but it is still part of the core functionality of the platform. Being able to hold a given position allows for easier testing and tuning of angle control, as well as any future features. As mentioned, this platform can find uses in education and research. In these cases, space is limited, and it is not desirable for the robot to drift away during testing.

- **Контрол на ъгъла** – Контрола на ъгъла е най-важната функционалност на Teensy BalanceBot Mk1. Контролирането на ъгъла не означава роботът да седи само в изправено положение. Помислете за случая, когато роботът трябва да се придвижи от точка А до точка Б. При преместването, поради естеството на робота, той трябва да се наклони или напред или назад. Това изисква повече внимание да се обърне по време на разработката, защото ъгъла на наклон не е константа и зависи от скоростта на робота.
- **Задържане на позиция** -Тази функционалност не е критична за робота, но е важна част от основната функционалност на робота. Възможността да се държи дадена позиция позволява за по-лесно настройване на контрол на ъгъла и тестване на бъдеща функционалност. Както беше споменато преди, тази платформа намира приложение в образованието и изследователската дейност. В тези случаи мястото за работа е ограничено и не е желателно робота да се измества по време на тестването.

Модел на системата

Създаването на модел на системата е критична стъпка от процеса на разработката. Той позволява потенциални проблеми да се открият на по-ранен етап, преди време и усилия да се вкарани в разработката на физически обект. Моделът също позволява да се направят симулации на системата. Тези симулации предоставят ценна информация, за това как системата се държи в реалния свят.

В случая на BalanceBot Mk1, симулациите дават възможност да се симулира как робота се контролира и как той реагира на даден физически импулс.

Commented [NT6]: Creating the model of a system is a crucial step in the development process. It allows for any problems to be found early before any time and materials have been put into physical prototypes. The model of a system also allows for simulations to be made. These simulations provide valuable insight into the way the system behaves in the real world. In the case of BalanceBot Mk1 simulations are also a great way to determine how the system can be controlled and how it will behave based on the input.

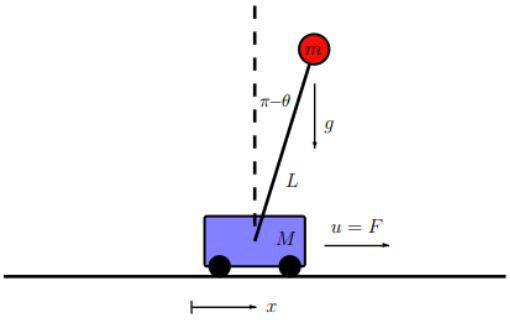
Математически модел

Създаването на математически модел е първата стъпка в създаването на цялостен модел на системата. Това е така, защото движението на всеки обект може да бъде описано със система от уравнения. Ако тези уравнения са достатъчно точни, симулацията също точно ще представи истинската системата.

Commented [NT7]: Creating a mathematical model is the first step of creating a model of the system. This is because the movement of body can be described by a set of equations. If the derived equations are accurate enough, the simulation will also be a true enough to the real system.

Модел с обърнато махало

Преди уравненията, които описват системата е важно системата да се разгледа в по-опростен вариант. В този случай е добре да се разгледа схемата на свободно тяло на фиг. 1. Вижда се, че системата може да се представи, като обърнато махало поставено на количка.



Фиг. 1 Схема на свободно тяло (Steven L. Brunton, 2017)

Колелата на робота могат да се представят като количката, а останалата част от тялото като махалото. Този модел не е свръхточен, но е достатъчно добър за проект.

Ако се изисква по-голяма точност, повече време трябва да се отдели на математическия модел.

Уравнения на движението

Уравненията на движението описват поведението на физическата система от гледна точка на нейното движение като функция на времето. (R.G. Lerner, 1991). Извеждането на тези уравнения от схемата на свободно тяло е трудоемък процес и извън обхвата на този проект, затова се използват уравнения, които вече са изведени

Commented [NT8]: Before the equations, the govern the system, it is important to look at the system in a simpler way. In this case as you can see in the FBD (Free Body Diagram) on (figure 1), the system is essentially an inverted pendulum on a cart.

Commented [NT9]: The wheels can be thought of as the cart, and the rest of the body is the pendulum itself. It is not a highly accurate model, but it is good enough for the goal of this project.
If higher accuracies are required, more time should be put into the modelling of the system.

Commented [NT10]: Equations of motion describe the behavior of a physical system in terms of its motion as a function of time (R.G. Lerner, 1991). Deriving such equations from a FBD is complicated and is outside the scope of this project, that is why this project uses equations of motion in figure 2 that are already derived.

$$\dot{x} = v \tag{8.67a}$$

$$\dot{v} = \frac{-m^2L^2g\cos(\theta)\sin(\theta) + mL^2(mL\omega^2\sin(\theta) - \delta v) + mL^2u}{mL^2(M + m(1 - \cos(\theta)^2))} \tag{8.67b}$$

$$\dot{\theta} = \omega \tag{8.67c}$$

$$\dot{\omega} = \frac{(m + M)mgL\sin(\theta) - mL\cos(\theta)(mL\omega^2\sin(\theta) - \delta v) + mL\cos(\theta)u}{mL^2(M + m(1 - \cos(\theta)^2))} \tag{8.67d}$$

Фиг. 2 – Уравнения на движението (Steven L. Brunton, 2017, p. 352)

Commented [NT11]: These non-linear equations of motion can be used to model the system and create a simulation. The details about creating the simulation are covered in subsequent sections.

Тези нелинейни уравнения на движението могат да се използват за моделиране на системата и създаването на симулация. Повече детайли за създаването на симулацията могат да бъдат намерени в следващите секции.

Възможни контролери

Commented [NT12]: When it comes to controllers there are a couple available options. This section is focused on examining three of the available options, analyzing pros and cons of each as well as ease of implementation.

Когато става въпрос за контролери, има няколко възможности. Тази секция се фокусира върху сравнението на трите възможности, анализирайки плюсовете и минусите им както и колко лесно е да се реализират.

Управляемост на системата

Commented [NT13]: This is a quintessential control theory problem. Before developing a system that requires control, it is important to check if it can be controlled. This check can be performed by calculating the rank of the controllability matrix of the system using the MATLAB command seen in Figure 3. In this case the rank should be 4 and if the code is run the rank will indeed be 4. This test confirms that the system can be controlled.

Това е типичен проблем в сферата на теорията за контрол. Преди да се разработи система, която се нуждае от някакъв вид контрол, важно е да се провери дали изобщо е възможно тя да се контролира. Тази проверка може да се направи като се изчисли рангът на матрицата на управляемостта използвайки MATLAB командата показана във фиг. 3.

В този случай рангът трябва да е 4 и при пускане на кода, той наистина извежда 4. Този тест потвърждава, че системата може да се контролира.

```
sys = ss(A, B, C, D);
ControlabilityMatrix = ctrb(sys);
rank(ControlabilityMatrix)
```

Фиг. 3 – ctrb() функция в MATLAB

Commented [NT14]: In figure 3 “sys” is a state-space representation of the system and the A & B matrices come from the equations of motion in figure 2, and the C and D matrices can be seen in figure 4.

Във фиг. 3 „sys” е представяна на системата в пространство от състояния и матриците A и B идват от уравненията на движението показани по-рано. Матриците C и D са показани на фиг. 4.

```
C = [1,1,1,1];
D = zeros(size(C,1), size(B,2));
```

Фиг. 4 - Матриците C и D

Commented [NT15]: The details about the state-space model, and mathematical theory are outside the scope of this documentation, but more information can be found in the book Data Driven Science & Engineering (Steven L. Brunton, 2017, p. 323).

Детайли свързани с модела в пространството от състояния и математическата теория са извън обхватата на тази документация, но повече информация може да бъде намерена в книгата “Data Driven Science & Engineering” *(Steven L. Brunton, 2017, p. 323)*.

Оптимален пълен контрол на състоянието – (ЛКР)/(LQR)

Commented [NT16]: Optimal full state control is accomplished using a Linear Quadratic Regulator (LQR). With this method, the eigenvalues of the closed-loop system (A − BK) are manipulated through choice of a full-state feedback control law u = −Kx (Steven L. Brunton, 2017, p. 343). For this method of control to be possible, the full state of the system must be available. This requires accurate sensors with little noise. Due to the available consumer-grade components discussed in future sections, this method is not a good choice for Teensy BalanceBot Mk1.

Оптимален пълен контрол на състоянието се осъществява с Линеен Квадратичен Регулатор (ЛКР) - (LQR) Linear Quadratic Regulator. С този метод собствените стойности на матрицата на системата със затворен кръг (A-BK) се манипулират с избор на закон за контрол на пълното състояние u=-Kx. *(Steven L. Brunton, 2017, p. 343)*.

За да се осъществи този метод, трябва да може да се измери пълното състояние на системата. Това изисква добри сензори, които имат малко шум в измерванията. Тъй като за този проект се използват компоненти (сензори) от потребителски клас този метод на контрол не е подходящ. Компонентите и сензорите се разглеждат по-късно в този документ.

Оптимален контрол с пълна оценка на състоянието (LQG)

Commented [NT17]: Full state estimation is usually accomplished with a Kalman filter. The full-state estimate from the Kalman filter is generally used in conjunction with the full-state feedback control law from LQR, resulting in optimal sensor-based feedback (Steven L. Brunton, 2017, p. 350). This method requires more analysis to determine how observable the system is. It’s possible that the full state of the system can not be calculated from certain measurements. This method also relies on good sensor data and that is why for the first iteration of this project it is not implemented.

Пълна оценка на състоянието се осъществява използвайки филтъра на Kalman (Калман). Пълната оценка на състоянието изчислена с помощта на този филтър се използва съвместно със закона от ЛКР и крайния резултат с оптимална сензорно базирана обратна връзка. (Steven L. Brunton, 2017, p. 350). С тази обратна връзка се осъществява контрол на системата. Този метод изиска повече анализ за да се установи колко е наблюдаема системата. Възможно е пълното състояние на системата да не може да бъде оценено при липсата на различни измервания. Този метод също зависи на добри сензорни данни и затова първата итерация на този проект не имплементира този начин на контрол.

ПИД (PID) Контролер

ПИД контролера е метод на контрол, който използва обратна връзка да контролира системи, които се нуждаят постоянно модулиран контрол. Този метод е широко разпространен в индустриални системи за контрол както и в други приложения като роботиката. Той е най-простия за имплементиране от трита метода разгледани в тази документация. Основното предимство на ПИД контролерите е, че по-лесно се имплементират на микроконтролер.

Подобен контролер не изисква сензори с голяма точност и дори и сензори с повече шум могат да се използват.

Поради по-лесната имплементация и олекотените изисквания за сензорите, Teensy BalanceBot Mk1 използва ПИД контролер.

ПИД контролера както и целия модел на системата са направени в Simulink, защото има всичките нужни инструменти за създаването на модела използвайки вече наличните 3Д модели. Повече информация за избора на софтуера има в следващата секция.

Софтуер за моделиране

Тази секция разглежда използвания софтуер за създаване на модел на системата. Това включва CAD софтуер, както и програми за моделиране на поведението и контрол на системата.

Commented [NT18]: PID Controller is a control loop mechanism that uses feedback that to control systems that require continuously modulated control. It is widely used in industrial control systems and a variety of other applications such as robotics. It is the simplest of the three control methods discussed to implement on a microcontroller. Such a controller does not require the highest accuracy from the sensors. Even noisy sensors can be good enough for a PID controller
Another benefit PID controllers is that unlike LQR and LQG, creating a model that uses a PID controller is significantly easier.
Because of the easier implementation and less strict sensor requirements, a PID controller will be used to control Teensy BalanceBot Mk1.
The PID controller along with the model are made with Simulink because it provides the tools necessary to create the model of a system based on existing CAD models. The creation of the model, controller and the simulation are covered in greater detail in the following sections.

Commented [NT19]: This section discusses the software used to model the system. This includes CAD software as well as software used for modelling the system behavior and control.
Modelling software is an important part of the development of any system. It provides a cheap way of creating the system in a virtual environment. This virtual twin of the system can be used for future manufacturing – for example, the 3D models designed in CAD can be used for 3D printing or machining. The 3D models can also be imported into software like MATLAB & Simulink to simulate the behavior and control of the system.

Софтуера за моделиране е важна част от разработката на която и да е система. Предоставя евтина начин да се създаде виртуална версия на системата. Този виртуален близнак, помага за бъдещето създаване на системата, например 3Д моделите разработени в CAD могат да се използват при 3Д принтирането. Същите 3Д модели могат да се вкарат в софтуер като MATLAB и Simulink за да се симулира поведението на системата.

Autodesk AutoCAD

CAD софтуера е най-важния от всички за този проект. Това е така, защото фазата на производство, така и фазата на моделиране в MATLAB зависят от 3Д моделите на системата.

Първата причина да се избера Autodesk AutoCAD беше, че проектантът, единствения човек работещ по този проект има опит с тази програма. Втората е, че тя има всичката функционалност, която проекта изисква.

Commented [NT20]: CAD software is the most important modelling software for this project, this is because both the manufacturing stage and the system modelling stage in MATLAB depend on having a 3D model of the system. The first reason Autodesk AutoCAD was chosen for this project as the sole designer of the system has experience working with the software. The second reason is that AutoCAD provides all the functionality required by the project.

MATLAB & Simulink

За създаването на модела на системата и да се моделира динамиката и контрола се използва MATLAB, Simulink и Simscape. Други продукти имат подобна функционалност, например Mathematica, но MATLAB има повече функционалност, специално направена за моделиране на системи.

Commented [NT21]: For creating the model of the system and to model the dynamics and control MATLAB with Simulink and Simscape is used. Other software products that have similar functionality exist such as Mathematica, but MATLAB is easier to use and has features designed specifically for system modelling.

MATLAB Модел и Симулация

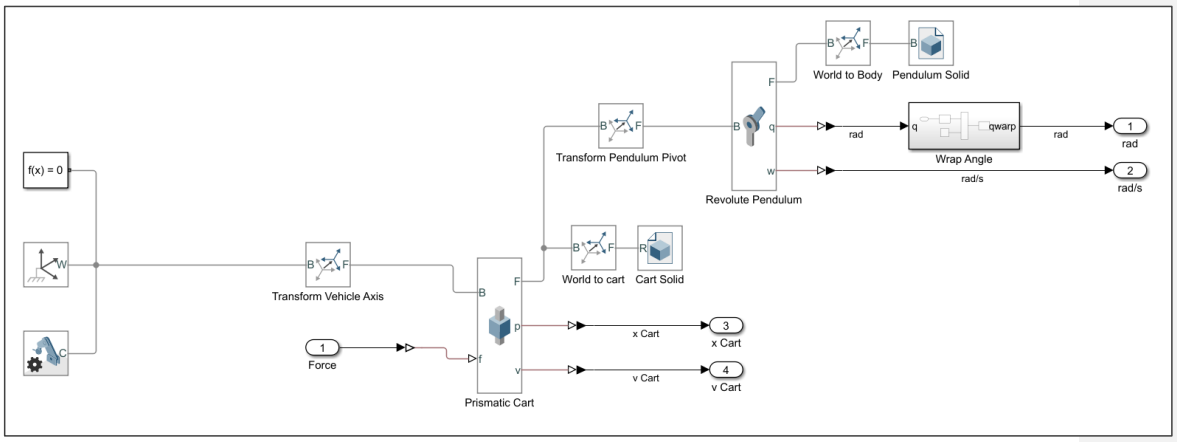
Тази секция обсъжда MATLAB модела създаден с помощта на Simulink, резултатите от него и разработения ПИД контролер.

Commented [NT22]: This section discusses the MATLAB model created in Simulink, the results from the model and the PID controller developed for the model.

Simulink Модел

Commented [NT23]: The Simulink model is divided into two parts. The first is the model of the robot, and the second are the PID controllers. The first part can be seen in figure 5, and the second in figure 6.

Simulink модела се раздела на две части. Първата част е модела на робота, а втората ПИД контролерите. Първата част е показана на фиг. 5, а втората на фиг. 6. По-големи версии на схемите могат да бъде намерени на Github (Nikola, 2021) страницата на проекта.



Фиг. 5 – Модел на робота (no control)

Модел на контролера

Тази секция разглежда модела на контролера какви входни данни приема, как е настроен и какви входни данни се приемат от модела на робота.

Модел на ПИД контролера

Както се вижда на фиг. 6, има 2 ПИД контролера. Вътрешния цикъл е за контрол на ъгъла, а външния е за задържане на позиция. И двата контролера използват вградения ПИД блок в Simulink.

Входните данни за контрола на ъгъла са съставени от две части – желания ъгъл и измерения ъгъл. Измерения ъгъл идва от революционната става на системата. Желания ъгъл може да се избере ръчно да е 0 или изхода на контролера за позицията.

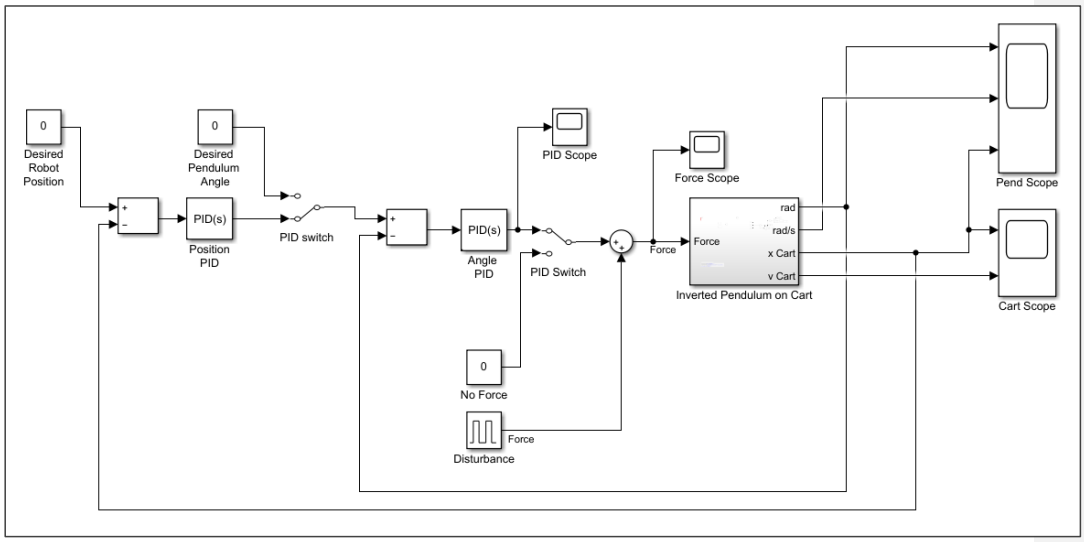
Commented [NT24]: This section looks at the controller model, what kind of input it takes, how it is tuned and how the input is fed into the robot model.

Commented [NT25]: As seen on figure 6, there are 2 PID controllers. The inner loop is for the angle control and the outer loop is for position control. Both use the PID block available in Simulink.

The input for the angle controller is comprised of two things – setpoint and actual angle. The angle measurement comes from the revolute joint of the system. The setpoint can be toggled either to 0 or to the output of the position control.

The position control takes in a setpoint which in this case is a constant block with a value of 0 and the position of the robot, which in this case is the position output from the prismatic joint of the model. The output, as mentioned before, is fed as the setpoint to the angle control.

Контрола за позицията приема желаната позиция, която в този случай е константен блок със стойност 0 и измерената позиция на робота, която се взима от призматичната става на модела. Изхода, както беше споменато преди се подава като желан ъгъл на контролера за ъгъла

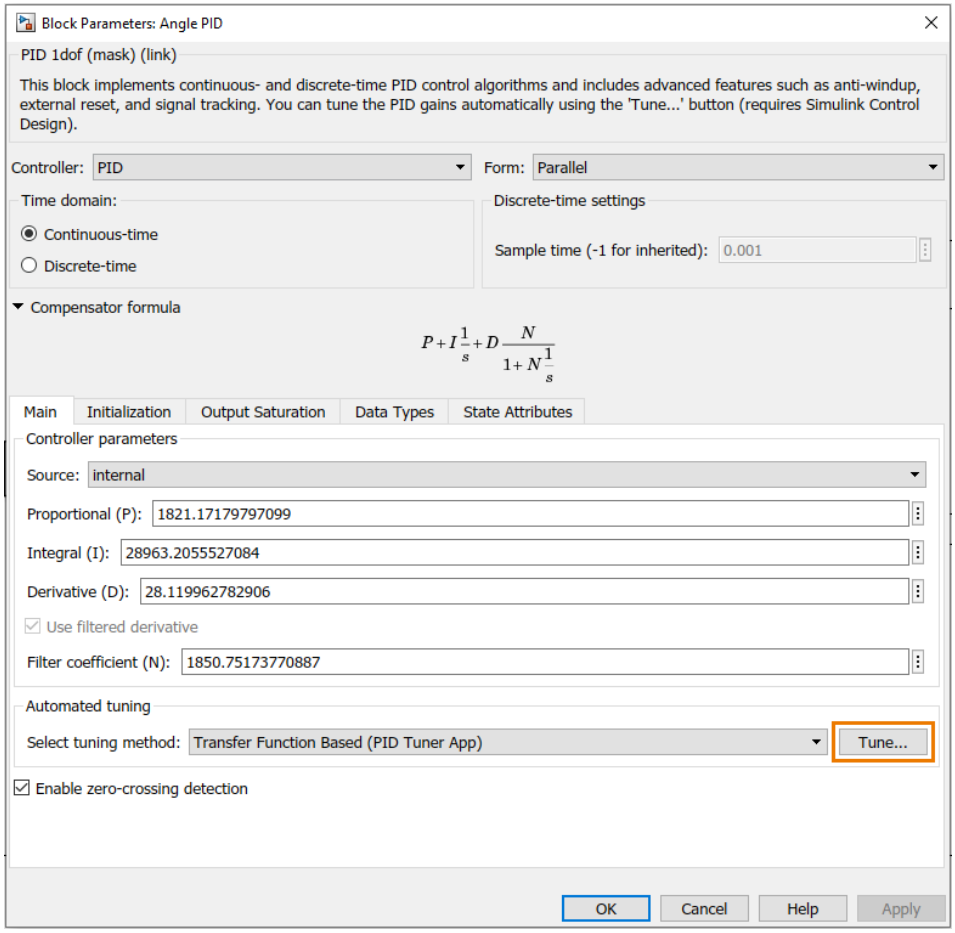


Фиг. 6 – ПИД контролери

Автоматична настройка на ПИД в MATLAB

Commented [NT26]: If the PID controller block is double-clicked, the property window opens (See figure 7).

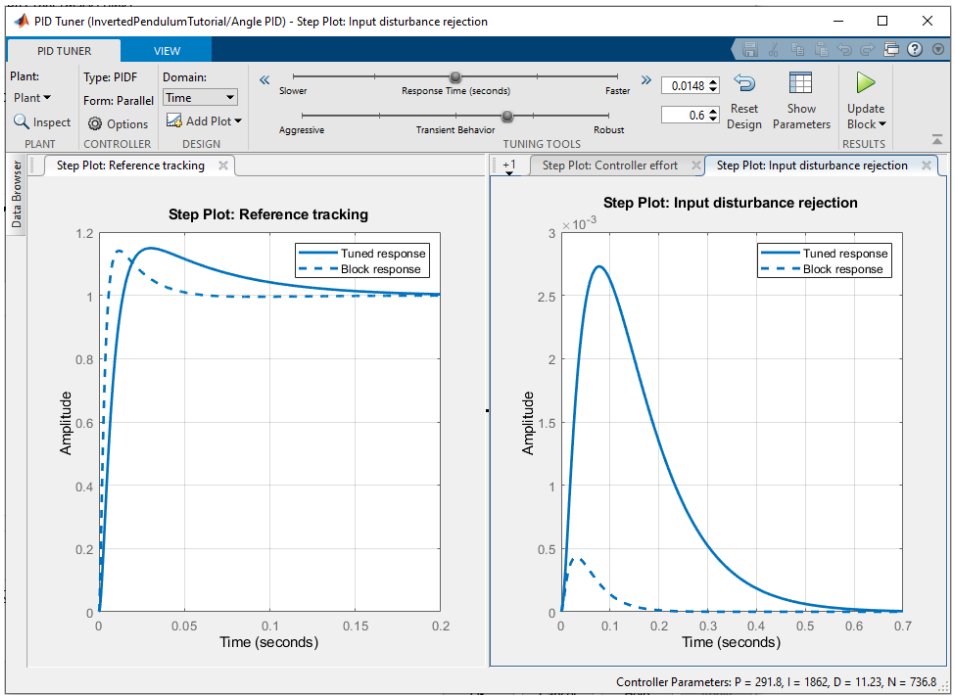
Ако болка на ПИД контролера се натисне два пъти, прозорец със свойствата на контролера се показва.



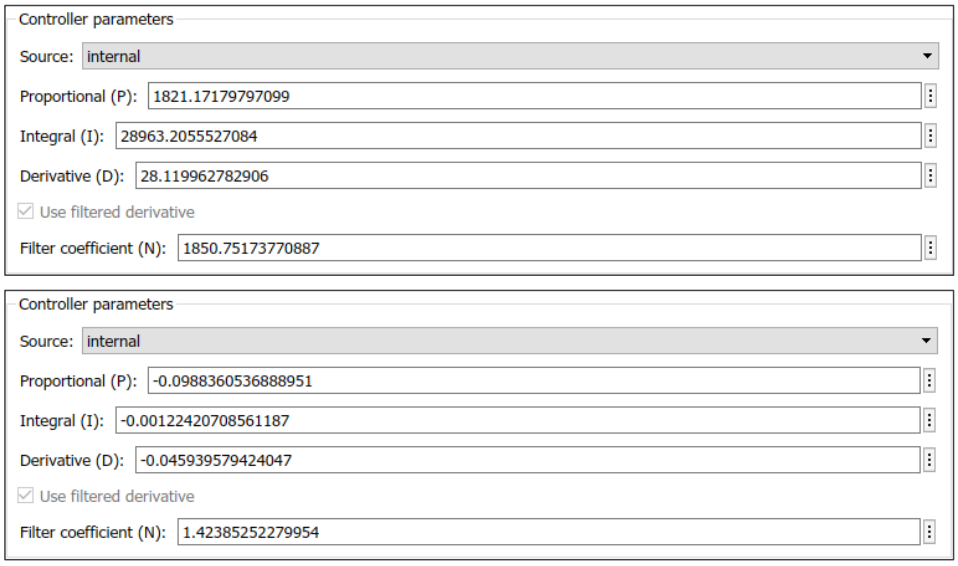
Фиг. 7 – Свойства на ПИД контролера

В долния десен ъгъл има бутон за настройване (tune button). Така автоматично се настройва ПИД контролера. Възможно е да се направи ръчна настройка, но е трудоемък процес и резултатите не са толкова точни.

След натискане на “Tune” бутона, нов прозорец се отваря (фиг. 8). От този прозорец различни параметри (колко бързо или бавно да реагира системата и колко да е стабилна) могат да се променят за да се промени поведението на системата. След тестване на различни стойности, на фиг. 9 могат да се видят крайните стойности, които се използват съответно за контролера за позицията и ъгъла.



Фиг. 8 – Прозорец за настройване



Фиг. 9 – ПИД за ъгъла (горе) и ПИД за позицията(долу)

Симулация

Симулацията се изпълнява в MATLAB и се визуализира като 3Д модел на робота. Преди симулирането на контролиран робот, се симулира без контрол за да се провери дали се държи по очакван начин. След като се установи, че поведението е правилно се включват ПИД контролерите и симулацията отново се пуска. Блок за смущения се добавя за да се симулира побутване на робота.

Резултатите от симулацията се записват като .mp4 файл.

Резултати от симулацията

Резултатите от симулацията могат да се видят в YouTube на следния линк:

https://youtu.be/9-L7qeeoDtc

или в Github страницата на проекта.

https://github.com/NikolaTotev/Teensy-Balance-Bot-Mk_1

Commented [NT27]: The simulation is run in MATLAB and is visualized as the 3D model of the robot. Before simulating control, the uncontrolled robot is simulated to verify that everything is working as expected. After everything has been verified to work correctly, the PID controllers are activated and the simulation is run again. A disturbance block is added to simulate the robot being pushed. The simulation results are recorded as an .mp4 file.

Commented [NT28]: The simulation results can be view on YouTube at the following link:
<INSERT YOUTUBE LINK>
or from the Github project page.
<INSERT GITHUB LINK>
The simulations show that without control the robot falls over and that when the PID controller is activated, the robot stays upright and keeps a position of 0. Any disturbances are also corrected.

Резултатите от симулацията показват, че без контрол, роботът пада и, че когато се активира ПИД контролера, роботът успява да се задържи във вертикална позиция и да задържа нулева позиция, а смущенията се компенсират.

Сравнение със съществуващи решения

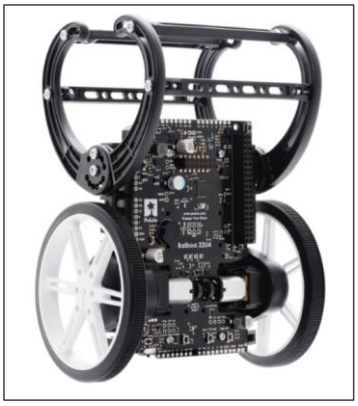
Тази секция анализира съществуващите решения, които са на пазара и имат подобна функционалност. Това сравнение ще вземе предвид както функционалността, така и възможните приложения, защото има много продукти, които имат подобна функционалност, но са предназначени за индустрията, имат много по-големи размери или и двете.

Commented [NT29]: This section analyzes current solutions on the market that have similar functionality. This comparison will consider the functionality as well as the use cases of the products. This is because there are products/projects that have related functionality, but they are used in industrial applications or are much larger or both.

Съществуващи решения

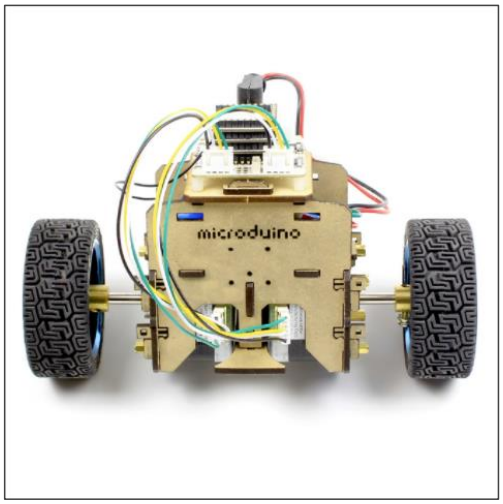
Тъй като има ограничен брой продукти, които имат подобна функционалност и подобни приложения, само два съществуващи продукта ще бъдат сравнени.*)*

- **Balboa 32U4 балансиращ се робот от Pololu –** Това е комплект, който може да се сглоби, като колелата и моторите трябва да се закупят допълнително. До края на сравнението, този продукт накратко ще бъде наричан Balboa. *(Снимка от* (Pololu, 2021)*)*



Commented [NT30]: Due to the limited number of products that have both similar functionality and use cases, only two solutions will be compared.
- **Balboa 32U4 Balancing Robot by** Pololu. This is a kit that can be assembled, wheels and motors need to be bought separately. During the rest of the comparison this will be referred to as ***Balboa*** for short. *(Photo from* (Pololu, 2021)*)*
- **Microduino Self-Balancing Robot by** Microduino. This is also a kit and comes with everything required. During the rest of the comparison this will be referred to as ***Mircoduino*** for short. *(Photo from* (Robotshop, 2021)*)*
-

- **Microduino балансиращ се робот от Microduino**. Това също е комплект, който се сглобява и идва с всичко необходимо. До края на сравнението, този продукт накратко ще бъде наричан Microduino. *(Снимка от* (Robotshop, 2021)*)*



Характеристики

Тази секция разглежда характеристиките на съществъващите решения и след това ги сравнява с Teensy BalanceBot Mk1 на база на списък от критерии които ще бъдат описани следващата подраздел.

Commented [NT31]: This section will look at the features each of the existing solutions have, and then they will be compared to the Teensy BalanceBot Mk1 based on a set of criteria that will be discussed in a following subsection. There is no mention of implemented functions, as both are kits and have to be programed by the end user.

Balboa

- Размер - 118 × 112 × 80 mm
- Тегло - 200g
- Микроконтролен - Arduino-compatible ATmega32U4
- Състои се само от една плакта.
- Вграден инерционен модул
- Вградени драйвери за мотори
- Вградени квадратични енкодери
- Интерфейс за Raspberry PI
- Използва четкови мотори
- 6 AA батерии

Microduino

- Размер - 19.5x8.5x11cm
- Тегло – 650 g
- Контролира се с Bluetooth
- Микроконтролер - Atmel ATmega1284P/ATmega644PA
- Радио модул: Microduino-nRF24
- Използва стъпкови мотори
- Две 3.7V Li-ion батерии.

Критерии за сравнение

От предходните списъци, двата екземпляра имат подобни характеристики, но има и разлики. За да се създаде равностойно сравнение между двата продукта и робота, който се разработва в рамките на този проект, ще се използват следните критерии:

- Модел на микроконтролера (MCU)
- Инерционен модул (IMU) и модел на модула (IMU Model)
- Тип използвани мотори (Motors)
- Тип драйвер за мотори (Motor Driver)
- Енкодери (Has encoders)
- Вид батерия (Battery Type)

Commented [NT32]: As the list shows, each robot has similar features, and some that are present in one but not the other. To create a fairer comparison all three robots will be compared with the following criteria:

- Microcontroller model
- IMU & IMU Model
- Motor type
- Motor driver
- Encoders
- Battery Type

Продукт	MCU Model	IMU & IMU Model	Motors	Motor Driver	Has Encoders	Battery Type
Balboa	ATmega32U4	Да - Неизвестен модел	Четков - 30:1 Micro Metal Gearmotor HPCB 6V	Вграден – Неизвестен модел	Има	AA – Презар. се
Microduino	ATmega1284P ATmega644P	Да – Неизвестен модел	Стъпков – Незнаен NEMA Модел	Вграден – Неизвестен модел	Няма	3.7V – Li-On
Teensy BalanceBot Mk1	Teensy 4.1 - ARM Cortex-M7	Да – Pololu LSM6DS33	Четков – Обикновен с предавки (неизвестна марка)	Външен – Cytron MDD3A	Има	3.7V – Li-On

Инструменти за софтуерна разработка

Тази секция накратко ще разгледа избора на инструменти за разработка на софтуера, тъй като е важна част от разработката.

Commented [NT33]: This section will briefly discuss the choice of software development tools as it is considered an important part of the project development.
The primary development software is Visual Studio with the Visual Micro add-on. This software was chosen over others such as Visual Studio Code and the Platform IO plugin or the Arduino IDE, because it offered more stability during development, as well as more features that aid in the development process.

Предимно се използва Visual Studio с приставката Visual Micro. Тази програма беше избрана пред други като Visual Studio Code с приставката Platform IO или Arduino IDE, защото предлага по-стабилна среда за разработка, както и повече функционалност, която улеснява процеса на разработка.

Разработка

Този раздел описва в детайли целия процес на разработката. Разделен е на две части – хардуер и софтуер. Всеки подраздел съдържа информация за детайлите взети предвид при разработката, всичко от избора на микроконтролер, сензори, актуатори до избора на материали, информация за сглобяване и разработка на софтуера

Commented [NT34]: This section describes the complete development process in detail. It is divided into two sections, hardware, and software. Each section contains all the details that were taken into account when developing the system, everything from microcontroller, sensor and actuator selection to material choices and assembly information.

Хардуер

Тази секция разглежда процеса на разработка на хардуера на BalanceBot Mk1.

Commented [NT35]: This section will discuss in detail the hardware development process of BalanceBot Mk1.

Микроконтролер

Избрания микроконтролер за Teensy BalanceBot Mk1, както името подсказва е Teensy 4.1. Този микроконтролер беше избран за проекта, заради следните характеристики:

- Има малък размери не изисква много място за инсталация, но въпреки малкия размер има много пинове и интерфейси, които могат да се използват за бъдещи допълнения, като светлини, екрани, актуатори или сензори.
- 32-битов процесор – Процесорът на Teensy 4.1 е 32-битов ARM Coretx M7 и работи на честота от 600Mhz и има много функционалност, най-важната от които е поддръжката на операции с плаваща запетая.
- Добра поддръжка – компанията, която произвежда Teensy 4.1 – PJRC има много добра поддръжка за продуктите си и имат активен форум, който помага при възникване на проблеми

Commented [NT36]: The microcontroller of choice for the Teensy BalanceBot Mk1 as the as the name suggests is the Teensy 4.1.
The Teensy was chosen as the MCU for this project, because of the following characteristics:
- It has a small formfactor and does not require a lot of space. However, despite the small form factor it still has a lot of pins and interfaces that can be used for any future additions, lights, screens, actuators and sensors.
- 32-bit processor - The processor on the Teensy is a 32-bit Arm Cortex M7, it runs at 600Mhz and has a lot of functionality, such as support for floating point operations.
- Good support – the company behind the Teensy, pjrc has very good support for their products.

Сензори

Тази секция обсъжда какви сензори се използват и защо.

Общ преглед и съображения

Тъй като BalanceBot Mk1 изисква само основна функционалност, само два сензора са необходими – инерционен модул (IMU) и енкодери за колелата. Поради ограничените ресурси, използваните сензори са от потребителски клас. Сравнително евтини са, но предоставят нужната функционалност.

Commented [NT37]: Due to the basic functionality required, the BalanceBot Mk1 needs only 2 sensors – and IMU and encoders for the wheels. Due to limited resources available for the development of the BalanceBot, the only available sensors are hobby grade ones. They are inexpensive and offer acceptable functionality.

IMU – Жироскоп и акселерометър

Инерционния модул, който се използва е LSMDS33 от компанията Pololu. Този инерционен модул има жироскоп и акселерометър, това означава, че с помощта на комплементарен филтър може да се осъществи sensor fusion.

Модула поддържа протоколите I2C и SPI. За този проект се използва I2C протоколът. Сензора има измерена стойност от 16 бита за всяка ос от жироскопа и акселерометъра. Предоставя допра точност за цената си и има малък размер, което позволява лесно да се монтира.

Commented [NT38]: The IMU being used in the BalanceBot is the LSM6DS33 breakout board from Pololu. This IMU module has a gyroscope and accelerometer, which means it can be used together with a complementary filter for sensor fusion.
It can use I2C or SPI, in the case of this project I2C is used. It has 16bit reading per axis for both the gyro and accelerometer. It provides good accuracy for the price and has a small form factor that allows it to fit almost anywhere.

Енкодери

Teensy BalanceBot Mk1 използва комплект магнитни енкодери от Pololu. Предназначени са да се използват с мотори, които имат удължена ос, но те бяха единствените енкодери достъпни по време на разработката. За да се използват правилно, специална част беше направена, която да държи платката на енкодера и адаптер беше направен да държи магнитите на оста на мотора. Повече информация относно дизайна и създаването на 3Д моделите може да се намери в раздела „CAD Модели“ Проблемите, които изникват заради начина на използване на енкодерите се разглеждат в раздела „Бъдещо развитие“

Commented [NT39]: The Teensy BalanceBot Mk1 uses the magnetic encoder pair kit from pololu. They are meant to be used on DC motors with an extended shaft, however they were they only encoders available at the time of development. To use them correctly, a special motor housing was 3D modelled and printed to mount the PCB of the encoder and a special shaft adapter was also 3D printed to mount to the output shaft of the geared DC motor being used. More information regarding the design and creation of the 3D printed parts can be found in the “CAD Design” section. The issues that come from using the encoders this way are discussed in greater detail in the section “Future Development”.

Актуатори

Актуаторите са от съществено значение за роботите и избора на правилния актуатор зависи от приложението. Тази секция обсъжда възможни актуатори, как се сравняват помежду си и защо един от тях е избран вместо другия.

Commented [NT40]: Actuators are essential for robots, and choosing the right actuator based on the use case is important. This section will discuss the actuator options for the BalanceBot Mk1, how they compare to one another and why one was chosen over the other.

Общ преглед и съображения

Тъй като BalanceBot Mk1 е мобилен робот, способността за точен контрол на въртенето на колелата е важна. Моторите трябва да имат и достатъчно мощност за да контролират робота. В случая на този робот, ако се използват четкови постояннотокови мотори, те трябва да имат предавки, защото малките мотори не са мощни. Друго нещо, което трябва да се вземе предвид е възможността за следене на завъртането на мотора в сравнение с зададената команда. Както беше споменато в предходни раздели, това се осъществява с магнитни енкодари. Това е още едно ограничение при избора на мотор.

Commented [NT41]: Since the BalanceBot Mk1 is a mobile robot, it is important to be able to accurately control the rotation of the wheels. The motors also need to have enough power to control the robot, in the case of the Teensy BalanceBot Mk1, this means that if DC motors are used, they must be geared, as small DC motors do not have a lot of power. Another important thing to consider is being able to read the actual position of the motor versus the commanded position. This is done through encoders, and as mentioned before in the sensors section, the BalanceBot Mk1, uses magnetic encoders, this adds further limitations to the choice of motor.

Мотори

Когато става въпрос за мотори, както беше споменато преди, една възможност е да се използват постояннотокови мотори с предавки, друга възможност е да се използват стъпкови мотори. Този проект използва постояннотокови мотори, заради енкодерите, които са достъпни. Друга причина да се използват постояннотокови мотори е мощност, стъпковите мотори са по-точни, но в повечето случаи не идват с предавки. По време на разработката на проекта, достъпните стъпкови мотори бяха по-слаби от постояннотоковите. Проблемите и възможни решения, които идват от използването на постояннотокови мотори се разглеждат в раздела „Бъдещо развитие“

Commented [NT42]: When it comes to motors, as mentioned, one option is to use geared DC motors, the other is to use stepper motors. For this project, DC motors are used, this is because mounting an encoder to a DC motor is much easier. Another reason for using DC motors is power, stepper motors are precise, but do not usually come with any kind of gears. The stepper motors available at the time of development, were not as strong as the DC motors and therefore despite the DC motors being more inaccurate, they were chosen as actuator for the BalanceBot Mk1. Issues, possible solutions and future upgrades regarding the actuators are discussed in the “Future Development” section of this document

Драйвер за мотор

Драйвера за мотор, който се използва в BalanceBot Mk1 е Cyrtron MDD3A. Може да работи с 4-16V, 3A и може да управлява два постояннотокови мотора или един стъпков мотор. Приема PWM сигнал като вход за определяне на скоростта и логическо ниско/високо ниво, за определяне на посоката. Драйвера има и вграден регулатор на волтажа, който доставя 5V.

Commented [NT43]: The motor driving being used in the BalanceBot Mk1 is the Cytron MDD3A driver. It has a maximum rating of 4-16V, 3A and can control 2 DC motors at once or 1 stepper motor. The input is PWM for speed and a high or low logic level for direction control. The motor driver has an integrated voltage regulator that supplies 5V.

CAD Проектиране и производство

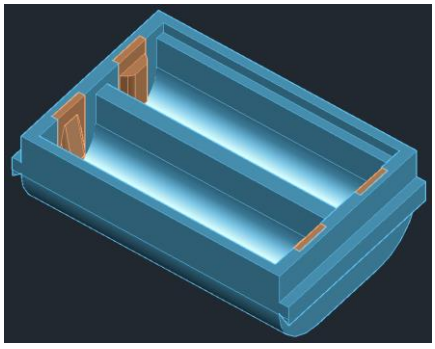
Този раздел разглежда структурата на робота, как е проектирана, основните части избора на материали и методи на производство.

CAD Дизайн

Тъй като BalanceBot Mk1 е съставен от малко на брой части, този подраздел ще ги разгледа по-подробно.

Отделение за батерията

Поставката за батерии се намира в долната част на робота, тя държи 2 Panasonic NCR18650PF Li-On презареждащи се батерии. Батериите се държат от две пружини направени от TPU покрити с медно тиксо за провеждане не ток.



Фиг. 10 – Отделение за батерията

Commented [NT44]: This section examines the main structure of the robot, how it was designed, what are the main sections of the robot, material choices and manufacturing methods.

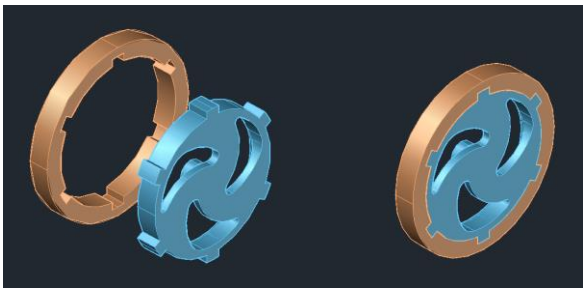
Commented [NT45]: Since the BalanceBot Mk1 has a small number of parts, in this section they will be discussed in some detail.

Commented [NT46]: The battery compartment is located at the bottom part of the robot, it houses 2 Panasonic NCR18650PF Li-On rechargeable batteries. The batteries are held in place by TPU springs with copper tape on the surface for conducting electricity. The electrical connections can be seen in the “Electronics” section.

Колела

Колелата са с диаметър 100мм. Съставени са от две части – твърда пластмасова главина, която се закача с триене към мотора и мека гума направена от TPU, която се захваща за главината с правоъгълни прорези. Специални колела бяха проектирани, защото тези, които идват с моторите са прекалено малки. Благодарение на меките гуми и здрава главина, тези колела работят като стандартните.

Commented [NT47]: The wheels are 100mm. They are comprised of two parts – a solid plastic hub that attaches with frictions and a keyed hole to the motor gearbox and a soft TPU tire that locks onto the hub with notches. Custom wheels were designed because the ones that came with the motors were too small. Thanks to the soft TPU and sturdy hub, these custom wheels perform as good as the default ones.

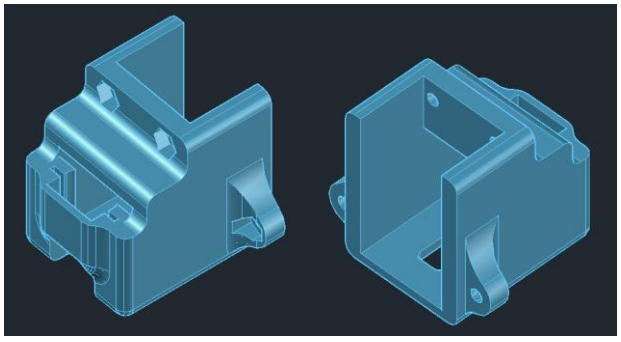


Фиг. 11 – Колела и гуми

Поставка за енкодерите

Поставките на енкодерите имат двойна функция, държат платката на енкодера на правилното разстояние от мотора и помагат за закрепването на моторите. Платката на енкодера е запоена за помощна платка и помощната платка влиза в жлеба на поставката. Има отвор за проводниците на енкодера. Няма нужда се допълнително закрепване на платките, защото триенето между частите ги държи достатъчно добре.

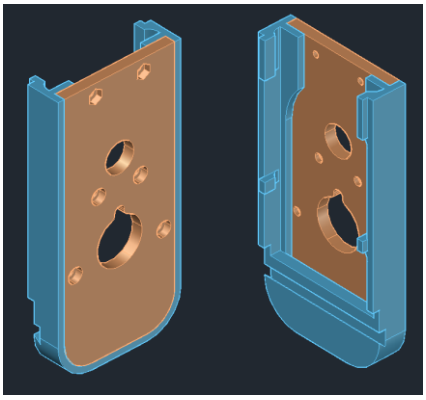
Commented [NT48]: The encoder mounts server two functions, they hold the encoder PCB at the right distance from the motor and they help to hold the motor secure. The encoder PCB is soldered onto another PCB, that slides into the slot of the mount. Below the encoder, there is a hole for the wires. There is no need for securing the PCB as it is a friction fit and because the robot never goes inverted.



Фиг. 12 – Поставка за енкодер

Долни странични панели

Долните странични панели държат моторите и поставката за енкодерите. Имат и трапецовидни релси, които служат за монтиране отделението за батерии. В горната част на панелите, скобите от средния панел се закачат.

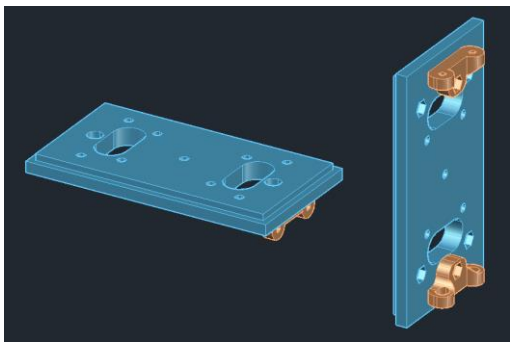


Фиг. 13 – Долни странични панели

Commented [NT49]: The bottom side panels hold the motors and encoder mounts. They also have dovetail rails, that are used for mounting the battery compartment. At the top of the bottom panels, a bracket attaches that is used for connecting the middle panel.

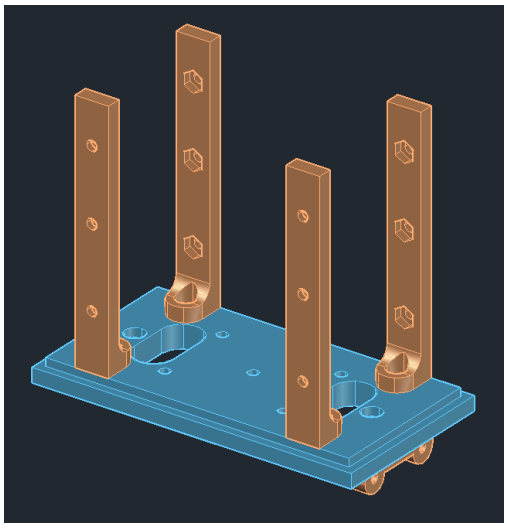
Среден панел

Средния панел е интерфейса между долната и горната част на робота. Има отвори за кабели и плоска част за монтиране на breadboard, който държи инерционния модул (IMU). Има два различни типа дупки, които служат за монтиране на крепежни скоби. Един тип за скобите да долната част на робота и втори за скобите на горната част на робота.



Фиг. 14 – Среден панел с долни скоби

Commented [NT50]: The middle panel is the interface between the bottom half and the top half. It has holes for cable management and room for a small breadboard that holds the IMU. There are two sets of mounting holes – one set for the bottom mounting holes and another for the vertical brackets that are used for mounting the top panels.

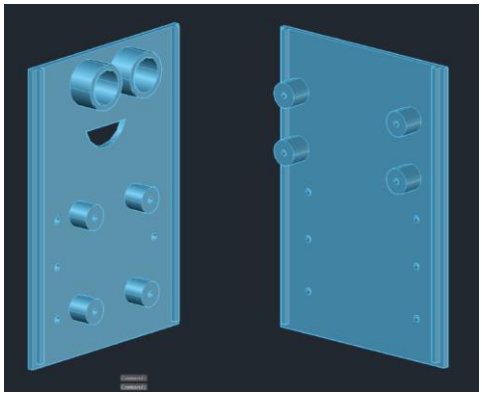


Фиг. 15 – Вертикални скоби

Горни панели (преден и заден)

Горните панели създават отсека за електрониката, който държи микроконтролера и драйвера за мотора. Има дупки за закачана на платките с дистанционни болтове. Горните панели, както беше споменато преди се закачат с вертикални скоби за средния панел (има две скоби за всеки) и 6 болта се използват за закрепването на панелите към скобите.

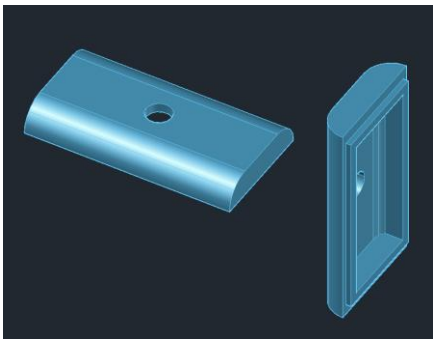
Commented [NT51]: The top panels create the electronics bay that contains the MCU and motor driver. The there are holes for mounting PCB standoffs to which the components are mounted to with small M3 screws. The top panels as mentioned before are attached to the middle plate with vertical brackets (two for each panel) and six screws are used for securing the panels to the brackets.



Фиг. 16 – Преден и заден горен панел

Горен капак

Горния капак се използва за монтиране на главния ключ за захранването, както и да добавяне на допълнителна тежест в горната част на робота



Фиг. 17 – Горен капак

Метод на производство

Метода на използван за производството на BalanceBot Mk1 е 3Д принтиране. Използва се, защото позволява за бързо създаване на прототипи и тестване на промени. То също е евтин и достъпен начин за създаване на сложни части.

Избор на материали

Материалите използвани за частите на BalancBot Mk1 са PLA и TPU.

PLA беше избран, защото по-лесно за принтиране от ABS. PLA е по-слаб от ABS, но този робот няма части, които изпитват големи натоварвания.

TPU с твърдост 70A на скалата на Shore се използва. TPU се използва предимно за гумите, защото те трябва да са меки да осигуряват сцепление. Този материал трудно се принтира, но с правилните настройки, резултатите са приемливи.

Commented [NT52]: The top cover is used for master power switch and for adding additional weight at the top of the robot.

Commented [NT53]: The manufacturing method for the BalanceBot Mk1 is 3D printing. It allows for rapid prototyping. It is also cheap for allows for making more complex parts.

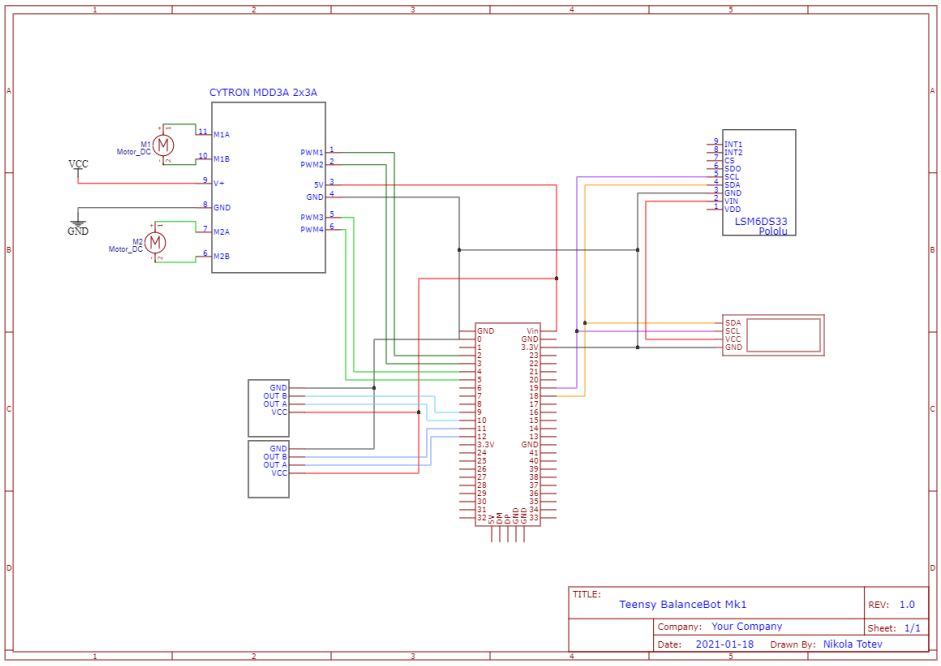
Commented [NT54]: The materials used for the structure of the BalanceBot are PLA and TPU. PLA was chosen because it is easier to print than ABS. PLA is weaker than ABS, but this robot does not have any demanding mechanical parts.
TPU with a hardness of 70A on the Shore scale. TPU is used only for the tires as they needed to be soft and provide grip. It is difficult to print, but with the right settings the results are good.

Електроника

Този раздел съдържа информация за електрониката на BalanceBot Mk1. Разглеждат се връзките между отделните компоненти и какви конектори и проводници за избрани

Диаграма на компонентите

Фиг. 18 показва диаграма на компонентите. По-голяма версия на схемата може да бъде намерена на Github (Nikola, 2021) страницата на проекта.



Фиг. 18 - System circuit diagram

Commented [NT55]: This section contains all of the information about the electronics of the BalanceBot Mk1. Connections between individual components are discussed, what connectors and wires are chosen. For more information about.

Commented [NT56]: Figure 18 shows the circuit diagram of the system. A larger version of the schematic can be found on the Github (Nikola, 2021) page of the project.

Монтиране на микроконтролера

Микроконтролера има пинове, които се вкарват в рейка. За по-лесна инсталация и деинсталация, както и улеснено тестване се използват двойни редове от рейки. Рейките за запоени за малка платка с размери 3x7см. Женски JST конектори са запоени за платката, те служат за лесно закачане на останалите компоненти за микроконтролера.

Commented [NT57]: The microcontroller has male pin headers that need to be plugged into female headers. To facilitate easy mounting a demounting of the MCU, as well as providing testing points for all of the pins, two double rows of headers were soldered onto a small 3x7cm prototype board. Female JST connecters were installed and wired to the correct pins to allow for easier connection of the other components of the robot.

Съединители и проводници

- **Съединители (конектори)**

Конекторите, които се използват в BalanceBot Mk1 за JST конектори с 2 и 4 пина, и Dupont конектори. Енкодерите и инерционния модул използват JST съединители, и мъжки JST към Dupont конектор се използва за връзката с драйвера на мотори.

JST конекторите осигуряват здрава връзка между компонентите, за разлика от Dupont съединителите, които лесно могат да се разделят. Единствената причина за използване на Dupont конектори е факта, че драйвер идва с такива.

Dupont конекторите също се използва за съединяването на breadboard върху който е монтиран инерционния модул. Проблемите, които възникват от този начин на инсталиране на IMU модула се разглеждат в раздела „Бъдещо развитие“

Commented [NT58]: The connectors used in the BalanceBot Mk1 are 2 and 4 pin JST connectors and Dupont connectors. JST connectors are used for the encoders and IMU, and a male JST connector is used for one side of the wires going to the driver board. Dupont connectors are used to connect to the driver board.
JST connectors provide a robust connection between components, and unlike the Dupont connectors they do not easily come apart by themselves. The only reason for using Dupont connectors is because the driver board comes with them, and it was not practical to de-solder them.
Dupont connectors are also used for connecting to the breadboard that the IMU is mounted to. The connection method for the IMU has flaws, these issues and possible solutions are discussed in the “Future Development Section.

- **Проводници**

Използваните проводници са многожилни и са със силиконова или PVC изолация. Препоръчително е изолацията да е само силиконова, защото е по-гъвкава и по-лесно се работи с нея в тесни пространства, но кабели с PVC изолация трябваше да се използват, защото Dupont конекторите идват с такива кабели.

В процеса на разработка, както се очакваше, кабелите със силиконова изолация бяха по-издръжливи и по-лесни за работа.

Commented [NT59]: The wires used are multistranded with either silicone or PVC insulation. The preferred insulation is silicone as it is very flexible and is easier to work with in tight spaces, however PVC needed to be used as it was the insulation that the Dupont connector cables came with.
During the development process, as expected the silicone wires were more durable and easier to handle.

Захранване

Батерии и поставка за батерии

Както беше споменато преди, отделението е направено от PLA и пружините за задържане на батериите са направени от TPU. Върху пружините е залепено медно тиксо и то е запоено серийно за. Отделението за батерии има мъжки JST конектор, който се закача за окабеляването закачено за главния ключ и драйвера за моторите.

Батериите, както беше споменато преди са 2 NCR18650PF 3.7V Li-On батерии с капацитет от 2700mAh

Commented [NT60]: As mentioned before the batter holder is 3d printed. The primary material is PLA and the springs that ensure the batteries are held firmly in place is made of flexible TPU. The batteries are wired in series as seen on the circuit diagram in the “Electronics” section. The battery holder has a male JST connector that hooks up to the wiring harness that is connected to the power switch and the motor driver.
The batteries as mentioned before are 2 NCR18650PF 3.7V Li-On batteries, each with a capacity of 2700 mAh.

Изисквания към захранването

Мощността, необходима за нормална работа, е около 0.8-0.9A при максимално натоварване.

Всеки мотор използва до 350mA.

Микроконтролера изисква 100mA и се захранва от 5V, които се предоставят от вградения в драйвера регулатор на волтажа.

Commented [NT61]: The power required for normal operation is around 0.8-0.9A at maximum draw.
Each motor draws up to 350mA.
The Teensy requires 100mA and is powered by 5V that is provided by the integrated power regulator on the motor driver board.

Сглобяване

Този раздел разглежда подробно детайлите относно сглобяването на робота. Избора на крепежни елементи и стратегии за сглобяване се обсъждат и сглобения робот е показан.

Commented [NT62]: This section will detail the final assembly details. Fastener choices and assembly strategies are discussed, and the final robot assembly is shown.

Избор на закрепващи елементи

Има много възможности за крепежни елементи. За размера на BalanceBot Mk1 и взимайки предвид толеранса на 3Д принтера, най-добрия избор са M3 болтове и гайки. Вдлъбнатина се създава в която влиза гайката и подобна се прави за главата на болта, за да не стърчи навън.

CA (cyranoacrylate) лепило се използва за залепяне на частите от страничния панел.

Commented [NT63]: There are many choices when it comes to fasteners. For the size of the BalanceBot 1 and taking into account the 3D printer tolerances, the best fastener choice are M3 nuts and bolts. A pocket is made to lock the nut in place, and one is also made to make the head of the blot be flush with the printed part.
CA (cyrano acrylate) glue is used to glue together the side panel assembly, because it is comprised of a frame and faceplate.

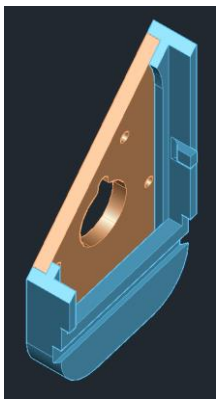
Стратегии за сглобяване

Commented [NT64]: As mentioned in the previous section, nuts and bolts are used. The primary assembly strategy is to use a central piece (the middle plate) to which brackets are attached, that then attach to the top or bottom pieces. This can be seen in figure 19.

Както беше споменато в преди, използват се болтове и гайки. Основната стратегия е да се използва централна част (средния панел), към него се закачат крепежни скоби, а върху тях се закрепят горната и долната част. Това може да бъде видяно на фиг. 19.

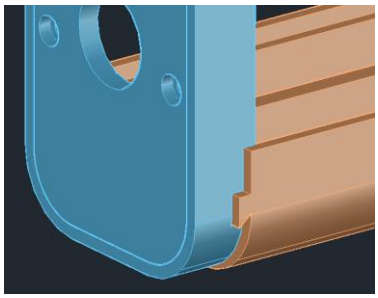
Фиг. 19 – Сглобка със скоби

Следващата стратегия може да бъде видяна на фиг. 20, малък ръб се прави на вътрешността на частта и върху не ляга външния панел и така се създава солидна плоскост. Когато се използват подобни ръбове, се добавя и CA лепило се по-голяма здравина.



Фиг. 20 – Сглобка с ръб

Последната стратегия се вижда на фиг. 21. Трапецовидна релса се използва, когато нещо трябва временно да се монтира, но лесно да се маха. В този случай се използва за отделението на батериите, тъй като трябва да може да се премахва, когато е време за презареждането им.



Фиг. 21 – Трапецовидни релси

Стъпки за сглобяване

Стъпка по стъпка инструкции могат да бъдат намерени на Github страницата на проект. Името на файла е AssemblyManual.pdf.

Commented [NT65]: Step by step assembly instructions can be found at the project Github repo. The name of the file is AssemblyManual.pdf

Сглобения робот

Фиг. 22 показва сглобения робот.

Commented [NT66]: Figure 22 shows the final assembled robot.

Фиг. 22 – Сглобения робот

Софтуер

Този раздел разглежда подробно процеса на разработка на софтуера на BalanceBot Mk1.

Commented [NT67]: This section will discuss in detail the software development process of BalanceBot Mk1.

Стратегия за разработка

Стратегията която се използва за разработка на софтуера е първо да се разработят отделните части и да се направи проверка дали работят правилно поотделно, след което да се съденият заедно. Различните компоненти се разглеждат в следващия раздел.

Commented [NT68]: The software development strategy used for the development of BalanceBot Mk1 was to design the different software components separately, test them to verify that they work and after that to combine all the working components. The software components are discussed in a subsequent section.

Софтуерна архитектура

Софтуерни модули

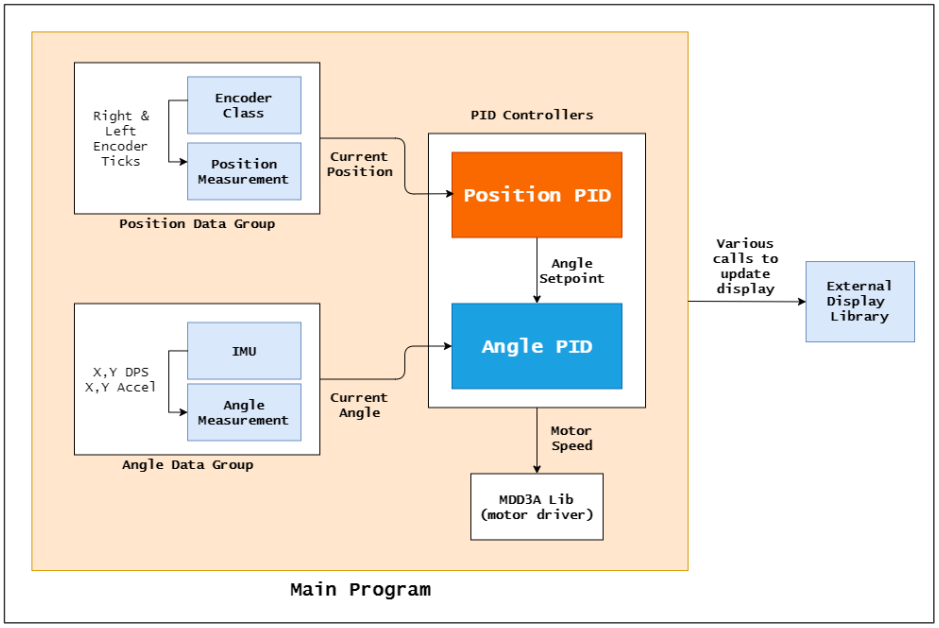
- **IMU –** Класът IMU управлява комуникацията с IMU модула. Предоставя функции, които връщат ъгловата скорост в градуса/секунда и ускорението за всички оси.
- **Encoder –** Класът Encoder съдържа кода необходим за прочитане на стойности от енкодера. Предоставя достъп до броя отброявания на енкодера както и посоката на въртене.
- **Position Measurement –** Класът Position се грижи да изчислението на позицията на база на измерванията от екнодерите. Предоставя методи, които връщат текущата позиция за всяко колело и Update метод, който се извиква от главната програма.
- **Angle Measurement –** Класът AngleMeasurement управлява изчислението и запаметяването на текущия ъгъл. Предоставя функции, които връщат текущия ъгъл на IMU модула за X и Y осите. Този клас трябва да бъде актуализиран в цикъла на главната програма
- **I2C Utilities –** Класът i2c_utils предоставя обвивка за функциите от библиотеката Wire.h. Тази обвивка е проектирана за по-лесно четене на и писане към регистри
- **Motor driver –** Класът MDD3A е интерфейса за драйвера на мотора. Предоставя единствено функцията UpdateSpeed(), която приема като аргументи скорост и посока.

Commented [NT69]: •**IMU –** The IMU class handles all communication with the IMU. It provides functions that output DPS and acceleration for all of the axis.
•**Encoder –** The encoder class takes care of the code required to read values from the encoder. It provides access to the number of ticks for each encoder as well as the last know direction.
•**Position Measurement –** The Position class handles the position calculation based on information from the encoders. It provides methods that return the current position for each encoder and an update method that must be called by the main loop of the program.
•**Angle Measurement –** The angle measurement class handles calculating and storing the current angle. It provides functions that return the current angle for the X and Y axis. This class must also be updated in the main loop of the program.
•**I2C Utilities –** i2c_utils class provides a wrapper for the functions from Wire.h. This wrapper is designed for easier reading and writing to sensor registers.
•**Motor driver –** This is class interfaces with the motor. It only provides one function – UpdateSpeed() and this function takes as arguments the speed and direction for the motor.

Окончателна структура

Фиг. 23 показва крайната структура на софтуера. По-голяма версия на схемата може да бъде намерена на Github (Nikola, 2021) страницата на проекта.

Commented [NT70]: Figure 23 shows the final software structure. . A larger version of the schematic can be found on the Github (Nikola, 2021) page of the project.



Фиг. 23 - Software Structure

Контролер

Настройка на PID

Както беше споменато в раздела „Автоматична настройка на ПИД в MATLAB“, програмата MATLAB има автоматична функция за настройване на ПИД контролери, въпреки това, тя не може дас е използва за ПИД имплементиран на микроконтролер. Поради тази причина, се използва емпирично настройване на ПИД контролера имплементиран на микроконтролера. Когато приемливо поведение на системата е постигнато, ПИД контролера се счита за настроен.

Commented [NT71]: As mentioned in the “Automatic PID Tuning in MATLAB” section, MATLAB has an automatic tuning function, however it cannot be used for tuning the implemented PID controller. Because of this limitation, the PID tuning was done empirically. Once an acceptable performance was reached the PID was considered tuned. The “Future Developments” section covers how this can be improved for future iterations of the project.

Commented [NT72]: If the PID controller block is double-clicked, the property window opens (See figure 7).

Внедрена функционалност

Режим на балансиране

Режима на балансиране е успешно имплементиран. Системата леко трепери, но това не се отразява негативно върху функционалността. Допълнително настройване на ПИД контролера може да помогне с премахването на този проблем, друго подобрение е по-добра инсталация на IMU модула. Тези подобрения се обсъждат в раздела „Бъдещо развитие“.

Commented [NT73]: Balance mode is successfully implemented. The system does “shiver” a bit, but that does not affect the balancing functionality. Additional PID tuning can help to reduce the shiver, but other fixes such as a soft IMU mount are required. These fixes and improvements are analyzed in the “Future Development” section.

Режим на задържане на позицията

Режимът за задържане на позицията също е имплементиран, но поради лоши сензорни данни не е точен колкото режима за балансиране. В раздела „Бъдещо развитие“ може да се намери повече информация за проблема и какви решения съществуват.

Commented [NT74]: Position hold is also implemented, however due to poor sensor performance it is not as accurate as the balance mode. The “Future Development” section has more information about the issues, and how they may be fixed.

Тестване

Тестването е ключова стъпка от процеса на разработка и този раздел ще разгледа тестовите стратегии, оборудване и резултати от тестовете

Стратегия за тестване

Тъй като повечето компоненти от Teensy BalanceBot Mk1 са свързани с съответен програмен код, всички тестове имат смес от хардуерни и софтуерни тестове.

Тестовата стратегия е да се разработи една функционалност, например кода за работа с IMU модула и след това дас е тества дали хардуера и софтуера работят правилно.

Тестването на малки части от системата осигурява, че когато всички компоненти се съберат заедно, те ще работят както трябва. Благодарение на тази стратегия минимален брой проблеми възникнаха при свързването на компоненеите.

Оборудване за тестване

Тестовото оборудване се състои от осцилоскоп, мултимер, настолно захранване и компютър.

Осцилоскопът се използва за проверка на сигналите които се предават между компонентите, както и да се провери дали дадени изчисления се изпълняват за правилното време.

Мултимерът се използва за проверка на електрическите вериги и запоените компоненти.

Компютъра се използва за качване на код върху микроконтролера, разработка на кода, и проверка на показанията от сензорите.

Commented [NT75]: Testing is a key step of the development process and this section will discuss testing strategies, test equipment and test results.

Commented [NT76]: Since most testable features for the Teensy BalanceBot Mk1 are tied to some kind of code, all tests have a mix between hardware and software tests.
The testing strategy used is to develop a single functionality –for example reading the IMU data and then testing if the hardware works and if the software is correctly reading and processing the data.
Testing small features often ensures that once all of the components come together, they will work as intended. Thanks to this testing strategy almost no issues were found when all of the components were combined into one system.

Commented [NT77]: The testing equipment used an oscilloscope, a multimeter, a bench power supply, and a computer.
The oscilloscope was used to check the type of signal being transmitted as well as for verifying that the system is working with the correct timing.
The multimeter was used in continuity mode to verify all of the solder connections.
The computer was used for uploading code to the MCU and verifying signal readings.

Описание на тестовете

Този раздел разглежда по-подробно проведените тестове.

Тест на електрическата верига

Тестовете на електрическата верига обхващат тестване на платките и окабеляването. Единствената платка, която трябва да се провери е тази, към която се закача микроконтролера.

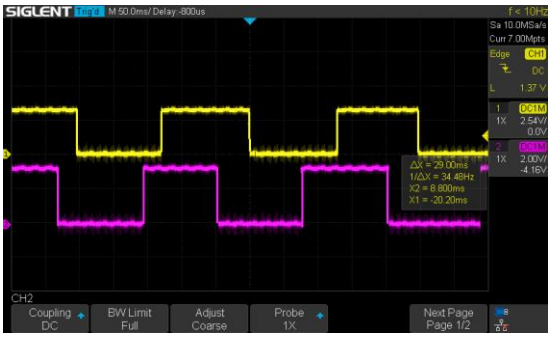
Проверката се прави с мултимера на режим „последователност“. Пробна жица се поставя в рейка, а друга се закача до крайната точка на дадения пин (например от пин за енкодера до платката на енкодера). По този начин се тества платката за микроконтролера, както и цялото окабеляване до сензора/актуатора.

Тест на енкодерите

Хардуерен тест

Теста за енкодера започва като се закачи енкодера за осцилоскопа и се захранва с 3.3V. Помощната платка се слага в поставката за енкодера и магнитите се поставят на моторите, а моторите се пускат да работят.

Ако енкодера работи правилно, на осцилоскопа трябва да се покаже квадратната вълна показана на фиг. 24.



Фиг. 24 - Encoder waveform

Commented [NT78]: This section will give a more detailed overview of the tests that were conducted.

Commented [NT79]: Circuit tests encompass PCB tests and wiring harness tests. The only PCB test that is required is the test for the MCU mount. The check is done with the multimeter in continuity mode. A lead is placed in the female header and another lead is attached to the end of the connection (for example from the PIN that handles encoder input to the pin of the encoder) by testing this way, the MCU mount is checked and at the same time the wiring harness for a given sensor/actuator is tested.

Commented [NT80]: The encoder test starts by connecting the wires attached to the encoder PBC to the oscilloscope and then powering it on. The power in this case comes from the motor driver which is powered by the benchtop power supply. The PCB is put into the encoder mount. The magnets are mounted to the motor and the motor is put into the encoder mount as well.

To test if the expected square waves are generated, the motor is turned on with the built-in buttons on the motor driver. If everything is working the encoder should produce the waveform seen on figure 24.

Софтуерен тест

Софтуерния тест се състои от закачане на енкодера за микроконтролера и преброяване завъртанията на енкодера с прост тестов код. Когато се установи, че кодът работи правилно, той се преработва и се превръща в модула за работа с енкодерите.

Commented [NT81]: The software test involves connecting the encoder to the MCU and then counting the ticks of the encoder with simple test code. Once this test code is verified it is improved and becomes the software module for working with the encoders.

Тестване на екрана

Тестът за екрана се състои от закачането на пиновете на модула за микроконтролера и използвайки външната библиотека от производителя да се тества дали модула работи.

Commented [NT82]: The screen test involves attaching the pins of the OLED screen to the power and i2c pins of the MCU and then using the library from the manufacturer to test if the screen works.

Тестване на инерционния модул IMU

Теста на инерционния модул се състои от закачането на правилните пинове на сензора към тези на микроконтролера. След това тестов код, който прочита ъглово ускорение и линейно ускорение. Ако този код работи и извежда достоверни резултати, се зарежда друг код, който използва данните от акселерометъра и жироскопа за да измери ъгъла на наклон на сензора. Сензора се накланя на ъгъл чийто размер се знае и се сравняват резултатите. Накланянето се прави с помощта на чертожен триъгълник.

Commented [NT83]: The IMU test consists of attaching the power and i2c wires and then using test code to read basic DPS and acceleration data. After that the angle measurement code is written, where the gyroscope and accelerometer data are combined. With the angle measurement code written the results are tested by attaching the IMU to a breadboard and then tilting the breadboard to a known angle. If the angle on the screen is the same as the known angle, the test passes.

Тестване на моторите и драйвера.

Моторите и драйвера за мотори се тества използвайки хардуерните бутони на драйвера. Когато се провери, че драйвера и моторите работят използвайки само бутоните, тестов код се пише за да се провери дали скоростта може да се управлява. Тестовата програма сменя скоростта от минимална на средна на висока и обратно до нула.

С осцилоскоп се проверява дали микроконтролера произвежда правилния PWM сигнал.

Commented [NT84]: The motors and motor driver are first tested by using the hardware buttons mounted to the driver board. Once its verified that the motors work and that the driver can control them, test code is written that controls the direction and speed. The output of that code is PWM wave and a high or low logic level. The output is check using an oscilloscope.
Once the output is checked, the MCU is connected to the motor driver and the code is run. The motors should cycle from minimum to maximum speed and then stop.

Бъдещо развитие

This section discusses some options for future development, what issues the current project have and how they can be solved.

Commented [NT85]: This section discusses some options for future development, what issues the current project have and how they can be solved.

Текущи проблеми

Неточни енкодери

Проблем

В момента, начина по който енкодерите са инсталирани, при едно пълно завъртане на колело, енкодера има 3 отброявания. Това означава, че точността е много лоша и има значително отместване на робота преди да има корекция.

Commented [NT86]: At the moment, the way the encoders are mounted, one wheel revolution equals only 3 encoder ticks. This means that the positional accuracy is very bad and significant position drift can be observed before any kind of correction is applied.

Решение

Има няколко решения на проблема, първото е да се инсталира по-добър енкодер (с повече стъпки на завъртане) за колелата. Друго решение е да се използват правилните мотори за енкодерите, такива с удължена ос отзад. На фиг. 25 се вижда правилното монтиране на енкодера и магнитите.

Commented [NT87]: There are several solutions, the first one would be to install better (meaning more steps per revolution) encoders to the wheels. Another one would be to mount the current encoders as the manufacturer intended. The encoders being used in the BalanceBot Mk1 are supposed to be used with DC motors that have an extended shaft at the back. As seen on figure 25 they are mounted to the rear of the motor and the magnets are mounted to the shaft.

Фиг. 25 - Correct encoder mounting

Монтиране на инерционния модул

Проблем

В момента IMU модула е закрепен за breadboard. Този метод създава няколко проблема. Първия е, че всички вибрации се долавят от сензора и създават шум в измерването, което се превръща в лошо поведение на системата. Друг проблем е, че модула лесно може да се наклони и да засича неправилен ъгъл, и това може да доведе до грешни команди към моторите. Третия е, че сензора е монтиран близо до драйвера за моторите, и заради честите промени на скоростта на моторите, това може да създаде електромагнитен шум, който да повлияе отрицателно на измерванията.

Commented [NT88]: Right now the IMU is mounted to a breadboard. This mounting method creates a couple of issues. The first one is that all of the vibrations are picked up by the sensor and they create a noisy measurement which in turn creates a poor control response. Another issue is that the IMU is not perfectly level and can easily shift. This causes an inaccurate reading which introduces a lot of instability. The third issue is that the sensor is mounted right next to the motor driver board, since it is working with higher currents and because the motor direction changes rapidly, this can cause a significant amount of electromagnetic interference that again can cause a noisy sensor reading.

Решение

Решението на първия и втория проблем е да се направи специална платка, върху която да се монтира модула. След това тази платка може да се монтира на платформа направена от TPU която да абсорбира част от вибрациите. Третия проблем може да се реши като се премести модула на по-подходящо място.

Commented [NT89]: The solution to the first problem would be to create a custom PCB (either printed or from prototype board) to mount the IMU to. This board can then be attached to a 3D printed TPU frame/platform that absorbs some of the vibrations.

Структура на роботи и сглобяване

Проблем

Сегашната структура на робота е прекалено малка за да се правят подобрения или да се добавят нови сензори.

Колелата са прекалено тънки и не предоставят достатъчно сцепление

Страничните панели се извиват при по-големи натоварвания.

Трудно се премахват батериите от отделението.

Няма добра организация на окабеляването, ако ще се добавят повече сензори, това трябва да промени.

Commented [NT90]: The current structure is too small for any kind of improvements. Its also difficult to assemble and maintain.
The wheels are too thin and do not provide enough grip.
The side panels twist if enough force is applied.
It is difficult to remove the batteries from the battery compartment.
There is no wire management. If more sensors are added, there has to be some kind of wire management solution.

Решение

За съжаление, за да се решат тези проблеми, роботът трябва да се проектира наново. Освен да реши сегашните проблеми, нова версия на робота ще позволи за правилно монтиране на нови сензори и актуатори.

Commented [NT91]: Unfortunately to solve the structural issues a complete redesign would be required. Along with solving the current issues a complete redesign would allow for correct mounting solutions for future sensor and actuator upgrades.

Мотори

Проблем

Моторите, които се използват не са с добро качеството и предавките имат много люфт. Точките за монтиране на колелата се са прави и се наблюдава клатене на колелата. Няма и начин за инсталиране на енкодери върху осите на мотора.

Commented [NT92]: The motors being used are not of very good quality, the gearboxes have a lot of backlash and the wheel mounts are not level and some wheel wobble can be observed. There is also no way to mount encoders to these motors.

Решение

Метални предавки, като тези предлагани от Pololu са по-добър вариант. Компанията Actobotics също произвежда добри мотори с вградени енкодери. Друга възможност е да се използват стъпкови мотори, но в този случай трябва да се добави втори драйвер или да се смени с друг модел, защото драйвера MDD3A може да контролира само един стъпков мотор.

Commented [NT93]: Metal gearboxes, such as the ones made by Pololu are a better option. Actobotics also make good motors that have integrated encoders. Another option might be to use stepper motors, but in that case a motor driver upgrade will be required, or at lease another motor driver needs to be purchased as the MDD3A can only control 1 stepper motor at a time.

Бъдещи подобрения

Този раздел ще разгледа някои възможни бъдещи подобрения, които мога да се направя за следващата итерация на робота.

Commented [NT94]: This section will explore some of the future improvements that can be made to the next iteration.

Хардуер

LiDAR Сензор

LiDAR сензор може да бъде добавен за точно измерване на разстояния. Този модул може да бъде инсталиран на въртяща се платформа за да се постигне 360 градусово зрително поле което да се използва в SLAM алгоритъм.

Commented [NT95]: A LiDAR sensor can be added for precise distance measurements. The LiDAR sensor can also be mounted on a rotating platform to provide a 360 field of “vison” that can be used for mapping the environment.

Сонар

Може да се добави сонар, който да се използва за избягване на колизии, защото има по-голямо зрително поле. Въпреки че има по-лоша точност от LiDAR модула може да се комбинират данните за да се направи карта на заобикалящата среда.

Commented [NT96]: A sonar sensor can be used for collision avoidance as it has a wider field of view. Even though it has a slightly less accurate distance measurement it can still be used together with the LiDAR module to map the environment.

Радиоуправление

Безжични модули за Bluetooth или WiFi могат да бъдат добавени за радиоуправление. LoRa модули могат също могат да се добавят и да се използват за телеметрия.

Commented [NT97]: Wireless modules, such as a Bluetooth or WiFi can be added to allow for remote control of the robot. A LoRa module can be added as well and it can be used for telemetry.

Функционалност

Създаване на графичен интерфейс

Графичен интерфейс може да се създаде за телефон или компютър. С него, роботът може да се настройва и контролира и може да се добави опция за жива видео връзка.

Commented [NT98]: A GUI interface can be created for a phone or computer. With it the robot can be controlled or reconfigured. A real-time video feed can also be added.

Добавяне на контрол за позицията

Към съществуващия код може да се добави такъв за контрол на посоката (напред, назад, ляв/десен завой). Командите за посоката могат да се изпращат от потребителския интерфейс или дистанционно.

Commented [NT99]: Code can be added that takes in direction instructions and the robot can be made to go forward, backward and turn. This can be done from a remote control or the GUI interface.

Избягване на препятствия

Може да се добави избягване на препятствия. Този режим може да се комбинира с контрол на позицията за да помага на управляващия да избягва сблъсъци с околната среда.

Commented [NT100]: Collision avoidance can be added, this can work together with the direction control to help the operator avoid obstacles.

Commented [NT101]: Collision avoidance can be added, this can work together with the direction control to help the operator avoid obstacles.

Картографиране на околната среда с помощта на ROS и SLAM

ROS (Robot Operating System) може да се инсталира върху Raspberry Pi, което да се монтира на робота. С помощта на ROS, SLAM (Simultaneous Localization and Mapping) алгоритъм може да се имплементира. С такава функционалност роботът може да стане напълно автономен.

Commented [NT102]: ROS(Robot Operating System) can be added to a raspberry pi on the robot. With it SLAM (Simultaneous Localization and Mapping) can be achieved. With this kind of functionality the robot can be made completely autonomous.

Източници

Nikola, T. (2021, 01 28). *Project Github Page*. Retrieved from Github: https://github.com/NikolaTotev/Teensy-Balance-Bot-Mk_1

Pololu. (2021, 1 18). *Pololu*. Retrieved from Pololu.com: https://www.pololu.com/product/3575

R.G. Lerner, G. T. (1991). *Encyclopaedia of Physics (second Edition)*. VHC Publishers.

Robotshop. (2021, 1 18). *Robotshop*. Retrieved from Robotshop: https://www.robotshop.com/en/microduino-self-balancing-robot-kit.html

Steven L. Brunton, J. N. (2017). *Data Driven Science & Engineering (Machine Learning, Dynamical Systems, and Control)*. Washington.