

Универзитет “Св. Кирил и Методиј” – Скопје
Факултет за информатички науки и компјутерско
инженерство

Проект по предметот Вградливи микропроцесорски
системи

Smart Cane – Automation for blind people

Михајло Крагујевски

183002

Предметен професор:

Д-р Моника Симјаноска

Содржина

Вовед	3
Фаза 1. Планирање и анализа	4
• Потреба (The Need)	4
• Целна група (Target Audience)	4
• Побарувања (Requirements)	4
• Конкуренција (Competitors)	4
Фаза 2: Дизајнирање	5
• Архитектура	5
• Технички Ресурси	6
• Алатки за развој	6
• Тим за развој	7
Фаза 3: Имплементација	7
Чекор 1.	7
Чекор 2.	7
Чекор 3.	8
Чекор 4.	8
Фаза 4: Продукција	10
Заклучок	11

Апстракт

Вид, способноста или моќта на гледање, приуштена на секој еден од нас. Не размислуваме на тоа, го правиме постојано, но што ако некој ја нема? Оштетување на видот, или уште познато како губење на вид, претставува намалена способност за гледање до степен што предизвикува проблеми што не можат да се поправат со вообичаени средства, како што се очила. Често дефинирано, од научен аспект, е коригирана острина на видот. Во 2015 год. околу 253 милиони луѓе биле означени со оштетен вид во светот. Од кои, 36 милиони биле целосно слепи, а останатите имале умерено до сериозно оштетување на видот. Ова е сериозна бројка на население со ваква препрека во животот. Со раст на технологијата на секое поле, се направени и разни достигнувања во оваа сфера: очила за процеп на светлина, принтери кои печатат хартија напишана со Брајова азбука и сл. За цел на овој проект, но и за старт на можеби нешто поголемо, одлучив и јас да додадам нешто, кое што мислам дека ќе биде од огромна помош за слепите лица а воедно ќе направи да размислиме малку повеќе за луѓето на кои им е потребна помош и како да ја искористиме технологијата во таа насока.

Вовед

Дали ги имате забележано лицата со оштетен вид?

Нивниот живот е полн со ризици. Тие не можат да се движат без помош на другите. Нивните животи секогаш зависат од други, па преку овој проект ќе ви ја доловам суштината и имплементацијата на мојата идеја.

Користејќи ја Arduino платформата, вклучена во програмата по предметот Вградливи микропроцесорски системи, се решив да направам Паметен стап за слепи лица (Smart Cane).

Со помош на оваа направа, лицето со оштетен вид ќе може да оди без ничија помош. Паметниот стап автоматски ја открива пречката пред лицето и му дава одговор на лицето во вид на вибрација и/или звук, како начин за предупредување. Преку ова, слепото лице ќе биде свесно за препреките пред него, и ќе може да ги одбегне.



Користејќи го Embedded Development Life Cycle (EDLC), ќе го прикажам општиот pipeline за развој на овој продукт. Почнувајќи од планирање и анализа, дизајнирање, имплементација и како за крај, продукција на финалниот производ.

Првата фаза се состои од објаснување за потребите, откривање која е целната група за користење на овој продукт, побарувањата кои ги имаат корисниците во врска со ваков тип на технологија и опис на конкуренцијата на пазарот.

Оваа фаза е битен дел од формирање на еден продукт бидејќи тука се гледа желбата и мотивацијата на инженерот, што го довело до тоа да размисли за ваква технологија и најважно се набљудува идеата од трета перспектива за тоа како оваа технологија ќе придонесе за подобрување на животот.

Фаза 1. Планирање и анализа

- **Потреба (The Need)**

Првично, овој процес за изградба на ваков продукт, почна како одговор на потреба. Лицата со оштетен вид имаат потреба од начини за тоа како да го олеснат начинот на кои тие функционираат во реалниот свет. Со губење на првичниот осет за вид, тие губат ориентација за простор, почнуваат да губат парчиња од меморија, а тоа е се поради оштетување на визуелниот кортекс, кој го користиме за сликовито помнење.

Потребата за ваква направа е за општа популација од индивидуален аспект, и е се со цел да им се даде помош на слепите лица во нивното секојдневно функционирање, како: движење по популаризирани улици, насочување во простор, предупредување за препреки кои може да преминат во несреќа и сл. Идеата ми е да се посвети повеќе внимание кон лицата со посебни потреби, и да се потенцираат истите, за да се направат продукти, наменети стриктно за нив.

- **Целна група (Target Audience)**

Откако потребата стана јасна, потребно е да ја дефинирам целната група. Крајните корисници на овој продукт се лица со оштетен вид (целосно или делумно слепи). Ова оштетување не бира пол, возраст или професија на лицето, така што тие атрибути не ги земам во предвид.

Од суштинско значење е да пренесам под кои услови би се користел продуктот. Придобивките од продуктот се самостојно движење на слепото лице. Нема да зависат од други луѓе или кучиња, посебни за таа намена, туку со овој уред ќе може без ничија помош да се движат и да се свесни за препреките околу нив.

- **Побарувања (Requirements)**

Секој вграден систем треба да решава некој проблем. Побарувањата во однос на функционалност и изглед, според фактот дека треба било кој да може да го користи, соодветува.

White cane, претставува бел стап кој што е симбол на ова попречување. Слепите лица го користат секојдневно за навигација низ средината. Бидејќи идеата ми е на веќе готов стап/држач, да прикачам коцка во која што ќе бидат сместени компонентите, јас само ја надградувам таа идеја и поради тоа продуктот нема да смени многу во начинот на кој што функционираат лицата со оштетен вид. Ќе биде генерално лесен, при што секој ќе може да се движи со него. Нема да биде потребен интернет или wifi конекција, така што може да се изнесува и надвор. Потребна ќе биде само батерија со чија помош ќе работи уредот, во случај да се испразни, потребно е друго лице да ја смени, и лицето со оштетен вид ќе може повторно да го користи паметниот стап.

- **Конкуренција (Competitors)**

Според мене, конкуренцијата не е толку изразена во оваа сфера, поради фактот што се гради продукт за подобрување на животот на други луѓе, но сепак постои. Моментот кога треба да го истакнам продуктот, да го рекламирам да достигне поголема популација, сето тоа е добивка за тоа што повеќе луѓе ќе слушнат за него толку повеќе и ќе го користат тие на кој што им е потребно. Согласно со ова, моментално не се грижам за конкуренцијата на пазарот, но ќе се посветам целосно на овој проект, за се со цел во иднина да се искористи идеата за да се надогради, и направи нешто многу подобро.

Фаза 2: Дизајнирање

После анализата и планската распределба на овој проект, следува фаза на дизајнирање, каде што ќе ја прикажам архитектурата - врз база на претходните чекори ќе ги дефинирам хардверот и софтверот во рамки на буџетот, технички ресурси – кои технологии ќе ги користам за развој на овој продукт, development kits и сл. И како за крај тимот за изработка со што ќе ја заокружам фазата за дизајнирање односно вториот чекор од EDLC циклусот.

- **Архитектура**

Главна, како што напоменав во претходната фаза, овој проект е врз база на Arduino, што значи и хардверскиот и софтверскиот дел ми се опфатени. Ќе користам Arduino UNO REV 3 табла, која што е најдобра за да се почне со електроника и кодирање. Бидејќи ова ми е прво искуство со ваква платформа, UNO е најробусната табла со која што може да почнам да играм. Таа е најкористената и најдокументирана табла од целото семејство на Arduino.

Arduino UNO е табла за микроконтролен базирана на ATmega328P(datasheet). Има 14 дигитални пинови за влез/излез (од кои 6 може да се користат како PWM излези), 6 аналогни влезови, керамички резонатор од 16 MHz, USB врска, приклучот за напојување, ICSP header и копче за ресетирање. Содржи се што е потребно за поддршка на микроконтролерот, едноставно се поврзува со компјутерот преку USB кабел или користејќи AC-DC



батерија за да се наполни. Оваа табла е многу корисна бидејќи може да си играме со неа како сакаме, без да се грижime премногу ако направиме нешто погрешно. Најлошо сценарио, го менуваме чипот и почнуваме пак одново.

“UNO” значи “еден” на италијански јазик и е избран за да го одбележи објавувањето на Arduino Software(IDE) 1.0 верзијата. UNO плочата е прва во низата на USB плочи на Arduino и исто така е референтен модел за платформата Arduino.

Покрај ова, го инсталирав open-source софтверот Arduino Software (IDE) кој што го олеснува пишувањето на код и неговото поврзување со плочата. Овој софтвер може да се користи со било која плоча на Arduino. Кодот се пишува во т.н sketches и откако ќе биде напишан се прави upload, во горната листа со копчиња, се компајлира кодот и се поврзува со Arduino плочата.

• Технички Ресурси

Целта на овој проект ми е да биде корисен исто така и лесен за да може да се направи. Генерално се состои само од неколку компоненти кои што се лесно достапни и прилично ефтини ако знаеме од каде да ги набавиме

Компоненти и материјали

- Arduino UNO плоча (Оваа плоча јас ја нарачав од aliexpress за одприлика 400ден, моментално уште чекам да ми стигне).
- Базер (Или некоја компонента која што може да испушта звук, овој дел и не е потребен многу, но за цел да биде пофункционален продуктот, би било добро да биде вклучено)
- Ултрасоничен сензор (сензор кој што ќе ми помогне да ја одредам далечината од стапот до некој објект во близина)
- DC мотор
- Jumper жици
- Некаков switch (прекидач или било што, што може да менува вредност on/off, слично како транзистор)
- 9V батерија
- 9V конектор за батеријата
- Arduino IDE

Голем дел од овие материјали се лесно набавливи на места каде има распродажба на стари технолошки делови и може да се набават за прилично ниска цена.

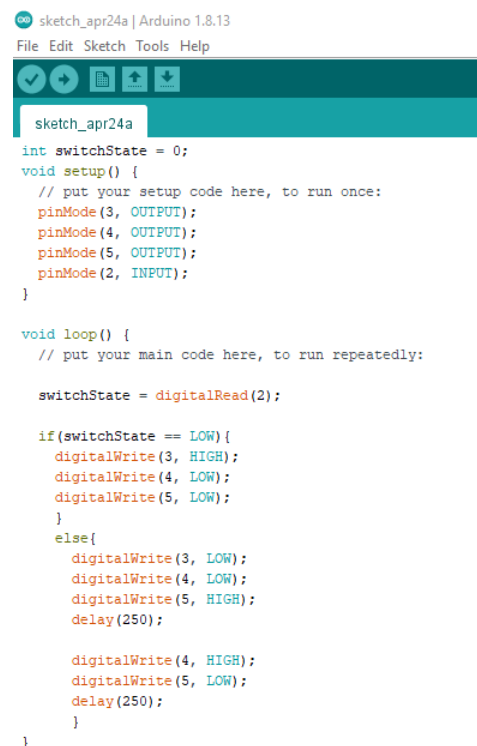
• Алатки за развој

Како што напоменав, ќе користам Arduino IDE. Овој IDE си има свој јазик за работа кој што е сличен на C++. Arduino е програмиран со дијалект од C/C++, што значи може да се користат и овие јазичи, но голем дел од библиотеките достапни од нив, не може да се користат. Овие рестрикции се направени поради ограничувањата заради малку достапната RAM меморија на хардверот Arduino.

Софтверот е прилично лесен за употреба, синтаксата е позната и распоредот за куцање е шематски кој се состои од два дела: setup и loop.

Овие се две методи кои што генерално во секој Arduino проект се состојат. Методата **void setup()** се користи да се иницијализираат променливите, се означуваат кои input/output пинови ќе се користат и сл. Оваа метода се извршува само еднаш.

Во методата **void loop()**, го ставаме главниот код, односно главната логика позади проектот и оваа метода се извршува постојано како во циклус. Работата на овие два методи може да се забележи на сликата од десно.



```
sketch_apr24a | Arduino 1.8.13
File Edit Sketch Tools Help

sketch_apr24a
int switchState = 0;
void setup() {
  // put your setup code here, to run once:
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(2, INPUT);
}

void loop() {
  // put your main code here, to run repeatedly:

  switchState = digitalRead(2);

  if(switchState == LOW){
    digitalWrite(3, HIGH);
    digitalWrite(4, LOW);
    digitalWrite(5, LOW);
  }
  else{
    digitalWrite(3, LOW);
    digitalWrite(4, LOW);
    digitalWrite(5, HIGH);
    delay(250);

    digitalWrite(4, HIGH);
    digitalWrite(5, LOW);
    delay(250);
  }
}
```

Пример код во Arduino

Може да си поставуваме дополнителни варијабли, константи и слично се со цел олеснување на пишувањето на кодот и полесно разбирање на логиката.

Во делот со tools има опција да се избере која плоча ќе ја користиме, бирам Arduino Uno, ја поврзувам плочата со USB кабел и имам хардверски и софтверски дел конфигуриран на една околина.

- **Тим за развој**

Проектите по предметот Вградливи микропроцесорски системи се изработуваат индивидуално, што значи овој проект е правен само од мене, без ничија помош. Од планирањето и анализата, па собирање на материјали, трошоци за истите, до искористување на ресурсите, имплементација на идеата и документација на целиот развој на проектот.

Фаза 3: Имплементација

Во оваа фаза на имплементација ќе објаснам како ги имам поврзано сите технички ресурси напоменати во втората фаза од овој проект, ќе ја прикажам шемата за поврзување на компонентите со Arduino и ќе ја објаснам структурата на кодот според кој се темели целата логика позади овој продукт.

Чекор 1.

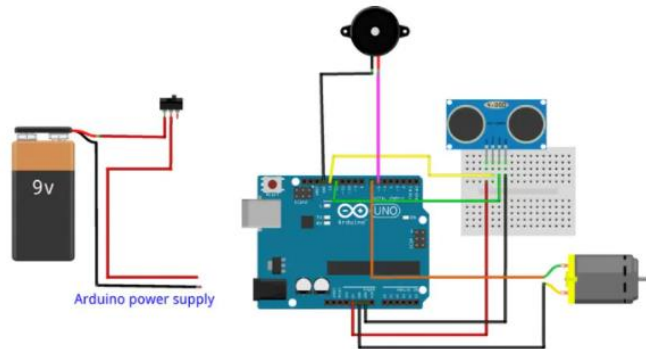
Најпрво, после собирање на сите потребни делови за реализација на оваа идеја, започнувам со добивање на најважната компонента, а тоа е базерот. “Базерот” може да биде било каква направа која предизвикува некаков сигнал до корисникот на стапот за да му означи дека има препрека пред него (за цел на оваа фаза ги земав двата вида во предвид и звук и вибрација). Тоа го добив со расклопување на стар мобилен телефон. Најдов расипан телефон, од постарите генерации, кој не е веќе употреблив, но неговиот микро – вибраторски мотор е сеуште во функција. Причината зошто го искористив ова е бидејќи е мал во големина и работи на низок напон. Откако ги расклопив сите делови на телефонот, од страната на куќиштето се наоѓа моторот на вибраторот, го извадив внимателно и го залемив на мало парче PCB за општа намета, потоа залемив две жици на терминалите на моторот. И со ова го добивам базерот.

Чекор 2.

Следно доаѓа ожичувањето на ардуиното. На шемата се прикажани сите делови поврзани со Arduino. Користев mini-breadboard за да го поврзам ултрасоничниот сензор со Arduino користејќи Jumper wires, додека па останатите делови како сигнализацијата и моторот се директно поврзани со Arduino.

Според наведениот дијаграм го добивам следното ожичување:

- Ultrasonic VCC to Arduino 5v
- Ultrasonic GND to Arduino GND
- Ultrasonic TRIG to Arduino D13
- Ultrasonic ECHO to Arduino D12
- Buzzer RED to Arduino D6
- Buzzer BLACK to Arduino GND
- Vibrator motor pin 1 to Arduino D7
- Vibrator motor pin 2 to Arduino GND
- 9 volt battery RED to Toggle switch pin 1
- 9 volt battery BLACK to DC male power jack(-)
- Toggle switch pin 2 to DC male power jack (+)



Со ова го завршувам жичаното поврзување на деловите.

Чекор 3.

За сите делови да бидат на едно место, одлучив да направам седиште од картонска кутија, за да може да го монтираам колото внатре. Вибрацискиот мотор го извадив надвор и го залепив користејќи изолациона лента се со цел да биде фиксиран за стапот. Моторот го залепив веднаш под рачката се со цел интензитетот од вибрацијата да биде појак и корисникот да може полесно да го осети.

Чекор 4.

И како за крај е поставување на sketch-от на Arduino. Најпрво ги дефинирам променливите за I/O и тоа ми се: trigPin, echoPin, motor, buzzer, иницијализирани соодветно според пиновите од плочата. Во setup() методата, ги наведов како да ми функционираат пиновите, trigPin, motor и buzzer како output пинови, додека на echoPin како input поради тоа што ми дава фидбек за пинот.

```
sourcecode $
#define trigPin 13
#define echoPin 12
#define motor 7
#define buzzer 6

void setup()
{
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(motor, OUTPUT);
  pinMode(buzzer, OUTPUT);
}
```

Во loop() методата иницијализирам две променливи и тоа duration – за тоа колку е времетраењето на пулсот од сензорот и distance – односно далечината од сензорот до објектот. Ги стартувам пиновите со delay од вкупно 12 microseconds, што е воопшто незабележително. Го поставувам echoPin на HIGH се со цел сензорот да ми го смета времето од почетокот на пулсот до крајот и функционира на тој начин што кога се приближува до предмет почнува да дејствува. Тоа го мерам со pulseIn() функцијата, која работи на тој начин што чита пулс (HIGH или LOW). pulseIn() чека пинот да премине од LOW во HIGH за да започне со тајмингот, потоа чека пинот да премине во LOW и го запира времето. Оваа функција ја враќа должината на пулсот во микросекунди, додека има пулс, или се откажува и враќа 0, ако не е добиен целосен пулс во тајмаутот. Вредноста ја внесувам во duration.


```

void loop()
{
    long duration, distance;
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);

    duration = pulseIn(echoPin, HIGH);
    distance = (duration/2) / 29.4;

    if (distance < 100) // Checking the distance
    {
        digitalWrite(motor,HIGH); // When the distance below 100cm
        digitalWrite(buzzer,HIGH);
    } else
    {
        digitalWrite(motor,LOW); // when greater than 100cm
        digitalWrite(buzzer,LOW);
    }
    delay(500);
}

```

За да ја пресметам далечината морам да го земам во предвид фактот дека работам со звук, односно звукот патува приближно 340 метри во секунда, ова одговара на одприлика 29,4 микросекунди на сантиметар. За да се измери растојанието што го поминал звукот, го земам моментот што звукот треба да патува напред и назад. Прво звукот се оддалечува од сензорот, а потоа отскокнува од површината и се враќа назад, односно далечината ја пресметувам по формулата $distance = (duration / 2) / 29.4$.

Следно морам да одберам минимална должина за сигнализација. Сметајќи дека корисникот кој го користи стапот, при негово движење, прави круг со радиус од 50cm (земајќи во предвид човекот да е во центарот, бидејќи тој управува), одлучив должината да биде барем 1m односно 100cm од крајот на стапот. Во овој случај, тие 50m растојание од човекот до крајот на стапот му се safe zone, каде што е сигурен дека нема препреки, додека па во наредните 100cm да може да биде известен доколку му следува некоја препрека пред него. За таа цел поставувам услов во случај distance да е помало од 100cm да ги постави моторот и базерот на HIGH за да почнат да сигнализираат, додека па ако растојанието е еднакво или поголемо од 100cm да ги стави на LOW односно да запрат. На крај од овој услов поставив delay() од 500 микросекунди односно од минатата проверка да се чека 0,0005 секунди за да може да се направи следната проверка и да се вклучи тригер пинот. Со ова ја завршувам логиката на кодот.

После сите овие чекори, тоа што преостанува е фазата на продукција, каде проектот е комплетиран и може да биде издаден на маркетот, да се прават разни анкети за тоа што мисли популацијата за продуктот, рекламација, deployment и слично.

Фаза 4: Продукција


Како финална фаза од овој проект претставува фазата за продукција на продуктот, односно продуктот е подготвен за масовно производство.

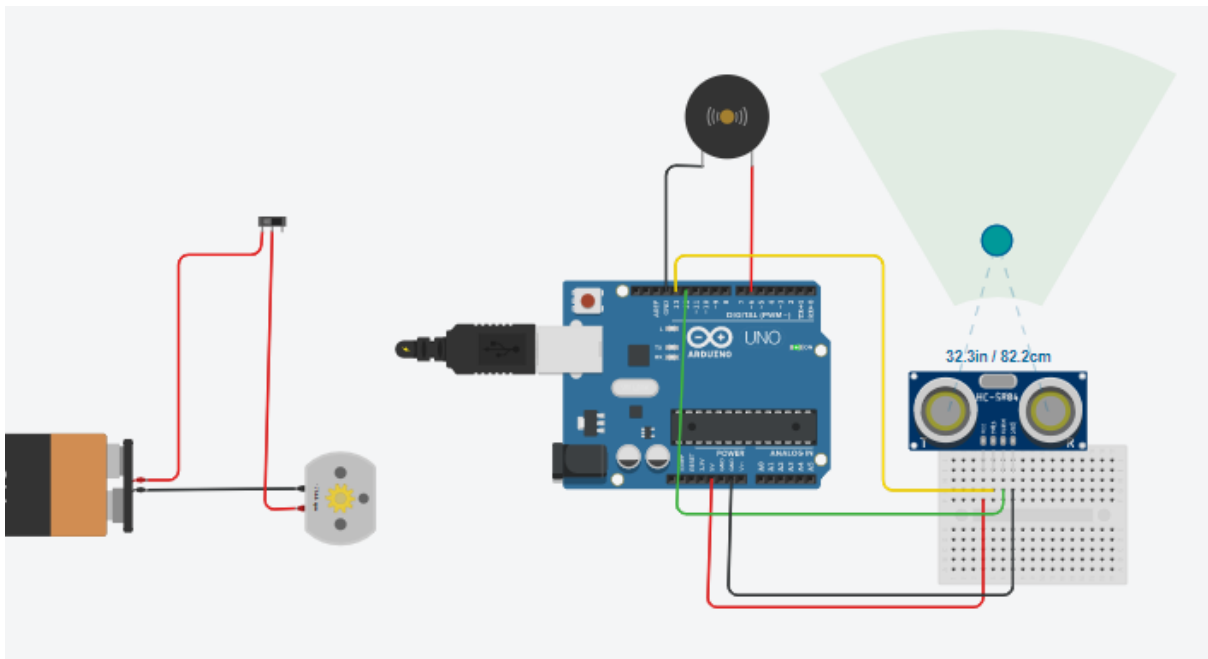
Пред да почнам со споделување на проектот, најпрво треба да го тестирам. За таа причина ќе го тестирам користејќи ја Tinkercad онлајн апликацијата за креирање електрични кола. Апликацијата содржи голем избор на компоненти за реализација на една идеја, како базери, мотори за вибрација, резистори, транзистори, LED светилки и сл, се со цел полесна репрезентација на реалниот продукт.

Линк до имплементацијата: <https://www.tinkercad.com/things/dd2204nD9In>

Имплементацијата ми е поставена како Public така што секој корисник може да пристапи до нејзе и да ја види.

Компонентите ги поврзав според објаснувањето во фаза 3, без вклучен вибрациски мотор. Тоа сепак е дополнителна компонента која служи за дополнителна сигнализација, меѓутоа звукот е доволен индикатор да покаже дека претстои пречка пред корисникот.

Со клик на копчето  **Simulate** добиваме дополнителни информации за симулација на проектот. Може да ја започнеме симулацијата, да се разгледа кодот и сл. Со стартување, времето започнува да тече и има е опција да раководиме со компонентите од кои зависи колото. Со клик на ултразвучниот сензор се појавува топче што е индикатор на некој објект, при што може да се забележи со негово поместување во ранг од 100cm од сензорот, базерот се вклучува и реагира дека има препрека.



Кога топчето ќе се измести од таа далечина, престанува да сигнализира базерот.

Со ова ја имам успешно тестирано имплементацијата, се што ми преостанува е самата продукција на идеата.

После оваа фаза, ќе ги вклучам сите документираните фази од проектот во една целина и ќе ги прикачам на GitHub. Бидејќи аспектот на овој проект е од наставно научно обележје, како проект по предметот Вградливи микропроцесорски системи, немам можност за дополнително маркетингање и рекламирање за по масовна продукција.

Линк до GitHub repository: https://github.com/KMihajlo/SmartCane_Automation

Заклучок

Користењето на технологијата за подобрување на човечки животи е тоа што не прави подобри луѓе. Со тоа развиваме по pametni имплементации, решаваме проблеми кои имаат клучен фактор врз нас, учиме нови работи и се развиваме на технолошки аспект како никогаш до сега. Се надевам и оваа имплементација ќе има некаков удел во градење на подобра иднина.